

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra elektromechaniky a výkonové elektroniky

BAKALÁŘSKÁ PRÁCE

Řídicí jednotka přestavníků pro modelovou železnici

**Autor práce: Luboš Boula
Vedoucí práce: Ing. Kamil Kosturik, Ph.D.**

2019/2020

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Luboš BOULA**
Osobní číslo: **E17B0198P**
Studijní program: **B2644 Aplikovaná elektrotechnika**
Studijní obor: **Aplikovaná elektrotechnika**
Téma práce: **Řídicí jednotka přestavníků pro modelovou železnici**
Zadávající katedra: **Katedra elektromechaniky a výkonové elektroniky**

Zásady pro vypracování

Navrhněte a realizujte řídicí jednotku přestavníků kompatibilní se stávajícím systémem modelové železnice.

1. Prostudujte způsob řízení a interní komunikace v rámci modelové železnice instalované na katedře aplikované elektroniky. Analyzujte možnosti ovládní přestavníků.
2. Navrhněte zapojení řídicí jednotky, která bude ovládat alespoň čtyři přestavníky. Jednotka musí umožňovat komunikaci po sběrnici CAN se zbytkem systému.
3. Navržené řešení realizujte.
4. Vytvořte příslušný firmware pro řídicí jednotku.
5. Detailně zdokumentujte jak vytvořený hardware, tak zejména funkce firmware včetně komunikačního protokolu.

Rozsah bakalářské práce: **30 – 40 stran**
Rozsah grafických prací: **podle doporučení vedoucího**
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

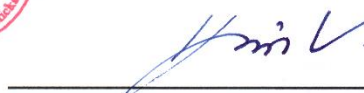
1. Vlaky TT [online]. [cit. 2019-04-10]. Dostupné z: http://projekty.fel.zcu.cz/index.php/Vlaky_TT.

Vedoucí bakalářské práce: **Ing. Kamil Kosturik, Ph.D.**
Katedra aplikované elektroniky a telekomunikací

Datum zadání bakalářské práce: **4. října 2019**
Termín odevzdání bakalářské práce: **11. června 2020**



Prof. Ing. Zdeněk Peroutka, Ph.D.
děkan



Prof. Ing. Václav Kůs, CSc.
vedoucí katedry

Anotace

Předkládaná bakalářská práce je zaměřena na řízení jednotek přestavníků na modelové železnici pomocí CAN komunikace. Návrh desky plošných spojů byl navržen v počítačovém programu Altium Designer 18. V práci je použitý mikrokontrolér MC9S08DZ60, který je naprogramován v jazyce C v počítačový programu CodeWarrior. V první části je popsán teoretický rozbor modelové železnice. Ve druhé části je blokové schéma pro řízení modelové železnice a jeho vylepšení. V další části je praktická realizace, která se zabývá návrhem desky plošných spojů, komunikačním protokolem a softwarem.

Klíčová slova

Komunikace CAN, programování v CodeWarrior, mikroprocesor MC9S08DZ60, program v C, Altium Designer, servomotory, přestavníky

Annotation

This bachelor thesis is focused on control of units of switch machines on model railways using CAN communication. The design of the printed circuit board was designed in the computer program Altium Designer 18. The work uses a microcontroller MC9S08DZ60, which is programmed in C language in the computer program CodeWarrior. The first part describes the theoretical analysis of the model railway. The second part is a block diagram for the control of the model railway and its improvement. The next part is a practical implementation that deals with the design of printed circuit boards, communication protocol and software.

Key words

CAN communication, programming in CodeWarrior, microprocessor MC9S08DZ60, program in C , Altium Designer, servo-motor, turnout

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....

podpis

V Plzni dne 14.6.2020

Luboš Boula

Poděkování

Na tomto místě bych rád poděkoval vedoucímu bakalářské práce, Ing. Kamilu Kosturikovi, Ph.D., za cenné připomínky, odborné a profesionální rady, kterými přispěl k vypracování této práce.

Dále bych chtěl poděkovat konzultantovi bakalářské práce, Ing. Jiřímu Žahourovi, Ph.D., za cenné připomínky, odborné a profesionální rady, kterými přispěl k vypracování této práce.

Dále bych rád poděkoval kolegovi, kterého si velmi vážím. Je jím kolega Patrik Albrecht. Patří mu mé poděkování především za rady a věcné připomínky při řešení projektu.

Také bych chtěl poděkovat celé mojí rodině a přátelům, který mě plně podporují ve studiu.

Obsah

OBSAH.....	8
ÚVOD.....	13
1 TEORETICKÝ ROZBOR MODELOVÉ ŽELEZNICE	14
1.1 MODELOVÉ KOLEJIŠTĚ A JEHO PROVEDENÍ.....	14
1.2 ZPŮSOBY ŘÍZENÍ MODELOVÉ ŽELEZNICE	15
1.2.1 <i>Digitální způsob řízení</i>	15
1.2.2 <i>Analogový způsob řízení</i>	15
1.3 PRINCIP SBĚRNICE CAN.....	15
1.3.1 <i>Komunikační protokol CAN</i>	15
1.3.2 <i>Síť sběrnice CAN</i>	16
1.3.3 <i>Recesivní a dominantní stav</i>	17
1.4 ZPŮSOB ŘÍZENÍ PŘESTAVNÍKŮ	18
1.4.1 <i>Odporový drát</i>	18
1.4.2 <i>Relé</i>	18
1.4.3 <i>Servomotory</i>	18
1.5 SERVOMOTOR	19
1.5.1 <i>Blokové schéma servomotoru</i>	19
1.5.2 <i>Reakce servomotorů na řídicí impuls</i>	20
1.5.3 <i>Řízení servomotorů</i>	20
2 KONCEPCE A BLOKOVÉ SCHÉMA ŘÍZENÍ KOLEJIŠTĚ	22
2.1 BLOK USB/CAN	22
2.2 BLOK CAN/DCC.....	22
2.3 BLOK DCC ZESILOVAČ	22
2.4 BLOK ŘÍZENÍ NÁVĚSTIDEL.....	23
2.5 BLOK ŘÍZENÍ PŘESTAVNÍKŮ	23
3 PRAKTICKÁ REALIZACE	24
3.1 POŽADAVKY NA ŘÍDICÍ JEDNOTKU PŘESTAVNÍKŮ	24
3.2 NÁVRH ZAPOJENÍ ŘÍDICÍ JEDNOTKY PŘESTAVNÍKŮ.....	24
3.2.1 <i>Tlačítko</i>	24
3.2.2 <i>RGB LED</i>	25
3.2.3 <i>Zdroj napětí</i>	26
3.2.4 <i>Mikrokontrolér</i>	27
3.2.5 <i>Obousměrný budič CAN sběrnice</i>	30
3.2.6 <i>Servomotory</i>	31
3.3 NÁVRH DPS.....	34
3.4 POSTUP NÁVRHU	34
4 KOMUNIKAČNÍ PROTOKOL	36
4.1 KOMUNIKACE PŘES CAN.....	36
4.1.1 <i>Zapojení CAN</i>	36
4.1.2 <i>Vytvoření CAN ID</i>	36
4.1.2.1 <i>Vytvoření ID pro řídicí jednotku přestavníků</i>	38
4.1.2.2 <i>Přiřazení CAN ID jednotce pro určitý typ dat</i>	38
4.2 ZPRÁVA PRO ŘÍDICÍ JEDNOTKU PŘESTAVNÍKŮ	39

4.2.1	Seznam funkcí řídicí jednotky přestavníků (výhybkového kontroléru)	39
4.2.2	Řízení servomotorů (manuální řízení)	39
4.2.3	Resetování řídicí jednotky přestavníků	40
4.2.4	Nastavení časové prodlevy mezi otáčením servomotoru	41
4.2.5	Zpětná vazba na vyžádání	42
4.2.6	Nastavení dorazů	43
4.2.6.1	Uložení dorazů	43
4.2.7	Výběr a přehození přestavníků (výhybek)	43
4.2.7.1	Přehození přestavníků (výhybek)	44
4.2.7.2	Výběr přestavníku (výhybky)	44
4.2.8	Nastavení čísla řídicí jednotky přestavníků	45
4.3	ZPRÁVA O NATOČENÍ SERVMOTORŮ PŘESTAVNÍKOVÉHO MIKROKONTROLÉRU	46
4.4	CHYBOVÉ HLÁŠENÍ PŘESTAVNÍKOVÉHO MIKROKONTROLÉRU	48
5	SOFTWARE	49
5.1	VÝVOJOVÝ DIAGRAM	49
6	ZÁVĚR	51
	SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	52
	PŘÍLOHY	1

Obr. 1.1 Modelová železnice [1]	14
Obr. 1.2 Síť sběrnice CAN a její impedanční přizpůsobení	16
Obr. 1.3 Rozdíl mezi diferenčním napětím a napětím proti zemi [8].....	17
Obr. 1.4 Blokové schéma servomotoru [3].....	19
Obr. 1.5 Natáčení servomotorů řídicími impulsy	20
Obr. 1.6 Řídicí impulsy pro čtyři servomotory	21
Obr. 1.7 Výstup kodéru pro čtyři servomotory	21
Obr. 2.1 Blokové schéma kolejiště	22
Obr. 3.1 Schéma zapojení tlačítek	25
Obr. 3.2 Schéma zapojení RGB LED.....	25
Obr. 3.3 Schéma zapojení spínaného step down regulátoru.....	26
Obr. 3.4 Schéma doporučeného zapojení [4]	27
Obr. 3.5 Schéma zapojení mikrokontroléru.....	28
Obr. 3.6 Schéma zapojení mikrokontroléru doporučené výrobcem [6]	29
Obr. 3.7 Schéma zapojení obousměrného budiče.....	30
Obr. 3.8 Použitý servomotor na řídicí jednotku přestavníků.....	32
Obr. 3.9 Schéma zapojení ovládní servomotoru.....	32
Obr. 3.10 Ukázka DPS spínaný zdroj LM2576S-5.0	34
Obr. 5.1 Vývojový diagram pro řídicí jednotku přestavníků	50

TAB. 3.1 ZAPOJENÍ MCP2551 PODLE VÝROBCE [5]	30
TAB. 4.1 PŘÍRAZENÍ ADRES KONKRÉTNÍM TYPŮM ŘÍDICÍCH JEDNOTEK	37
TAB. 4.3 VYTVOŘENÍ CAN ID	38
TAB. 4.4 PŘÍKLAD POŽADAVKŮ	38
TAB. 4.5 PŘÍRAZENÍ ID JEDNOTLIVÝM TYPŮM ZPRÁV	38
TAB. 4.6 ZPRÁVA PRO NASTAVENÍ ŘÍDICÍ JEDNOTKY	39
TAB. 4.7 KOMPLETNÍ SEZNAM FUNKCÍ	39
TAB. 4.8 ŘÍZENÍ JEDNOTLIVÝCH SERVOMOTORŮ	40
TAB. 4.9 RESETOVÁNÍ JEDNOTKY	40
TAB. 4.10 NASTAVENÍ ČASU MEZI OTÁČENÍM SERVOMOTORU	41
TAB. 4.11 ZPĚTNÁ VAZBA PRO ZJIŠTĚNÍ POZICE SERVOMOTORŮ	42
TAB. 4.12 POPIS FUNKCE PRO VYBRÁNÍ TYPU ZPRÁVY	42
TAB. 4.13 NASTAVENÍ DORAZŮ DOLEVA A DOPRAVA	43
TAB. 4.14 ULOŽENÍ	43
TAB. 4.15 VÝBĚR PŘESTAVNÍKŮ	44
TAB. 4.16 PŘEHOZENÍ DOLEVA ČI DOPRAVA	44
TAB. 4.18 NASTAVENÍ ČÍSLA JEDNOTKY	45
TAB. 4.19 ZPRÁVA O NATOČENÍ SERVOMOTORŮ VPRAVO, VLEVO A MEZI... ..	46
TAB. 4.20 ZPRÁVA O NATOČENÍ SERVOMOTORŮ VE STUPNÍCH	46
TAB. 4.21 PRO FUNKCI 0X01	47
TAB. 4.22 PRO FUNKCI 0X02	47
TAB. 4.23 CHYBOVÉ HLÁŠENÍ	48
TAB. 4.24 SIGNALIZACE RGB LED	48

Seznam symbolů a zkratek

CAN.....	Kontrolér oblasti sítí (Controller Area Network)
DPS.....	Deska plošných spojů
C.....	Programovací jazyk C
DCC.....	Digitální ovládání příkazů (Digital Command Control)
CNC.....	Numerické řízení počítače (Computer Numerical Control)
RC.....	Rádiové řízení (Radio Controlled)
CAN_H.....	Vysoké diferenční napětí (Controller Area Network High)
CAN_L.....	Nízké diferenční napětí (Controller Area Network Low)
MKO.....	Monostabilní klopný obvod
GND.....	Uzemění (Ground)
USB.....	Univerzální sériová sběrnice (Universal Serial Bus)
PC.....	Počítač (Personal Computer)
AD.....	Převodník z analogového na digitální
UTP.....	Nestíněná sériová sběrnice (Universal Serial Bus)
LED.....	Luminiscenční dioda (Light Emitting Diode)
RGB.....	Červená, zelená a modrá barva (Red, Green a Blue)
PTC.....	Termistor (Positive Thermal Coefficient)
NXP.....	Firma (NXP Semiconductors)
Rx_CAN.....	Pin v mikrokontroléru
Tx_CAN.....	Pin v mikrokontroléru
PWM.....	Pulzní šířková modulace (Pulse Width Modulation)
MOSFET.....	Unipolární tranzistor (Metal Oxide Semiconductor Field Effect Transistor)
THT.....	Konstrukce elektronické součástky (Through Hole Technology)
SMD.....	Konstrukce elektronické součástky (Surface Mount Technology)
ID.....	Identifikátor
BOM.....	Seznam součástek (Bill of Material)
EEPROM.....	Elektronicky vymazatelná paměť (Electrically Erasable Programmable Read Only Memory)

Úvod

Práce se zabývá řídicí jednotkou přestavníků pro modelovou železnici, která bude komunikovat po sběrnici CAN. V tomto projektu použiji servomotory, jako řídicí prvky modelového kolejiště. Snahou není pouze řídit přehazování výhybek, ale také vyhodnocování chyb, které nastanou při komunikaci s řídicí jednotkou přestavníků. Dále budeme schopni nastavovat dorazy servomotorů, a dokonce budeme schopni řídit servomotory manuálně.

Práce obsahuje detailní hardwarový návrh desky plošných spojů a rozebrání klíčových zapojení. Dále komunikační protokol pro přehlednost funkcí. V práci je rozebrán i příslušný firmware. Snahou je kompletní text rozdělit do logických a ucelených bloků pro snadnější orientaci v textu. První část se zabývá obecným popisem způsobu řešení a problematikou. V druhé části je detailně popsán hardware a v poslední části je popsán software.

1 Teoretický rozbor modelové železnice

1.1 Modelové kolejiště a jeho provedení

Modelové kolejiště je v měřítku TT, to znamená, že rozchod kolejnice je 12 mm a odpovídá to měřítku 1:120. Modelová železnice má tři stanice. Tyto stanice se nazývají Lhota, Beroun a Karlštejn. Lhota má jednokolejnou vedlejší trať, která má zabezpečení automatickým stavědlem do koncového nádraží. Dále sem spadá i testovací okruh. Beroun má hodně kolejí pro manipulaci s vlaky. Obsahuje i depo pro vlaky. Karlštejn poskytuje omezený nákladní provoz.

Modelové kolejiště má přibližně 300 kolejových úseků. Napájení vlaků a jejich řízení je provedeno pomocí kontaktu dvojkolí s vodivými kolejnicemi, které jsou od sebe izolovány. Jednotlivé úseky jsou od sebe taky navzájem elektricky izolovány. Úseky jsou od sebe co nejvhodněji oddělené, aby bylo možné zjistit, na kterém úseku se lokomotiva nachází. Každá souprava, odebírá elektrický proud. Proudový odběr způsobuje dekodér v lokomotivě a zatěžovací odpory ve dvojkolí vagónů. Na modelovém kolejišti, je možno ovládat návěstidla, výhybky a točnu. [2]



Obr. 1.1 Modelová železnice [1]

1.2 Způsoby řízení modelové železnice

Mezi způsoby řízení modelové železnice patří řízení: Digitální a Analogové.

1.2.1 Digitální způsob řízení

Využíváme konstantní amplitudu napětí. Kódování informace spočívá v době trvání mezi změnami polarity napětí v kolejnicích (DCC signál). DCC signál generuje řídicí jednotka na úrovni 5 V. Signál je třeba zesílit, protože po usměrnění napájí připojené zařízení. Těmito zařízeními jsou například vlaky, výhybky, návěstidla a další. Každé zařízení obsahuje dekodér, který čte z DCC paketu příslušnou adresu. Pokud dojde ke shodě adresy v DCC paketu a adresy dekodéru, pak se provede akce podle obsahu datového bajtu. Dekodér je realizován jako jednočipový mikro počítač.

Signál se šíří pomocí dvou kolejnic. Úseky kolejí nemusí být mezi sebou izolovány. Pomocí adresování je zajištěn provoz více vlaků na jednom kolejovém úseku. Nejsme ale schopni určit přesnou polohu soupravy na kolejišti. [2]

1.2.2 Analogový způsob řízení

Do kolejnic je přiváděno stejnosměrné napětí. Každá lokomotiva má jako pohon stejnosměrný motor napájený tímto napětím. Zde můžeme regulovat otáčky motoru, tedy rychlost lokomotivy. S rostoucím napětím roste rychlost lokomotivy. Změna směru tzn. že budeme provozovat opačný směr, je třeba změnit polaritu napětí pro dané kolejiště.

Jelikož analogové řízení se nepoužívá pro řízení přestavníků, tak se tímto způsobem řízení nebudeme dále zabývat. [2]

1.3 Princip sběrnice CAN

1.3.1 Komunikační protokol CAN

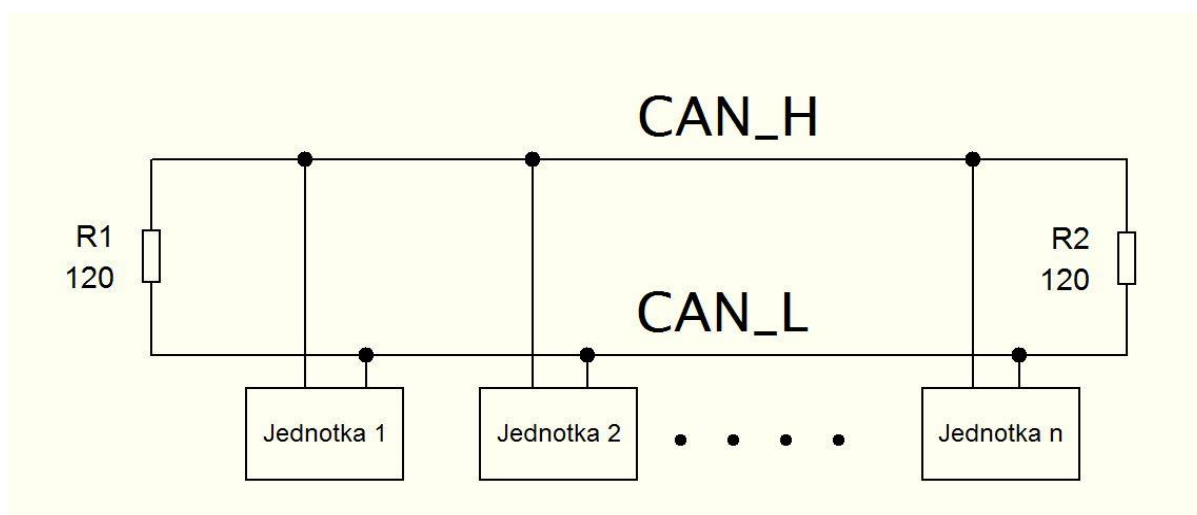
Controller Area Network zkráceně CAN je komunikační protokol, který byl vyvinut pro automobilový průmysl. V našem případě mluvíme o standardním formátu CAN 2.0 B, což je sériový komunikační protokol, který se využívá ke komunikaci jednotek po sběrnici. Tento protokol je spolehlivý, a ne příliš složitý, ba i dokonce je dosti levný. Další hlavním cílem bylo, aby odolával elektromagnetickému rušení, dokázal propojit vysoké množství

periferií a zároveň poskytoval dostatečně vysokou přenosovou rychlost. Komunikace je velice jednoduchá, spočívá v posílání krátkých zpráv obsahujících nula až osm datových bajtů. Tyto zprávy může posílat jakákoliv periferie. Při vyslání zprávy jsou všechny periferie v klidovém režimu (tedy mlčí), vyslanou zprávu buď to přijímají nebo jí zcela ignorují. Takovéto komunikaci se říká multi-master. Prakticky zde mohou periferie komunikovat a zároveň se řídit. [8]

1.3.2 Síť sběrnice CAN

Cílem sběrnice CAN je to, že musí zajistit realizaci logického součinu. Co to znamená? To znamená, že se na sběrnici vyskytuje v klidu stav recesivní, to je logická jednička, a pak kdykoliv a kterákoliv periferie vysílá dominantní bit, to je logická nula. Tento stav se již ovlivnit nedá.

Na Obr. 1.2 je znázorněna síť sběrnice CAN a její impedanční přizpůsobení.

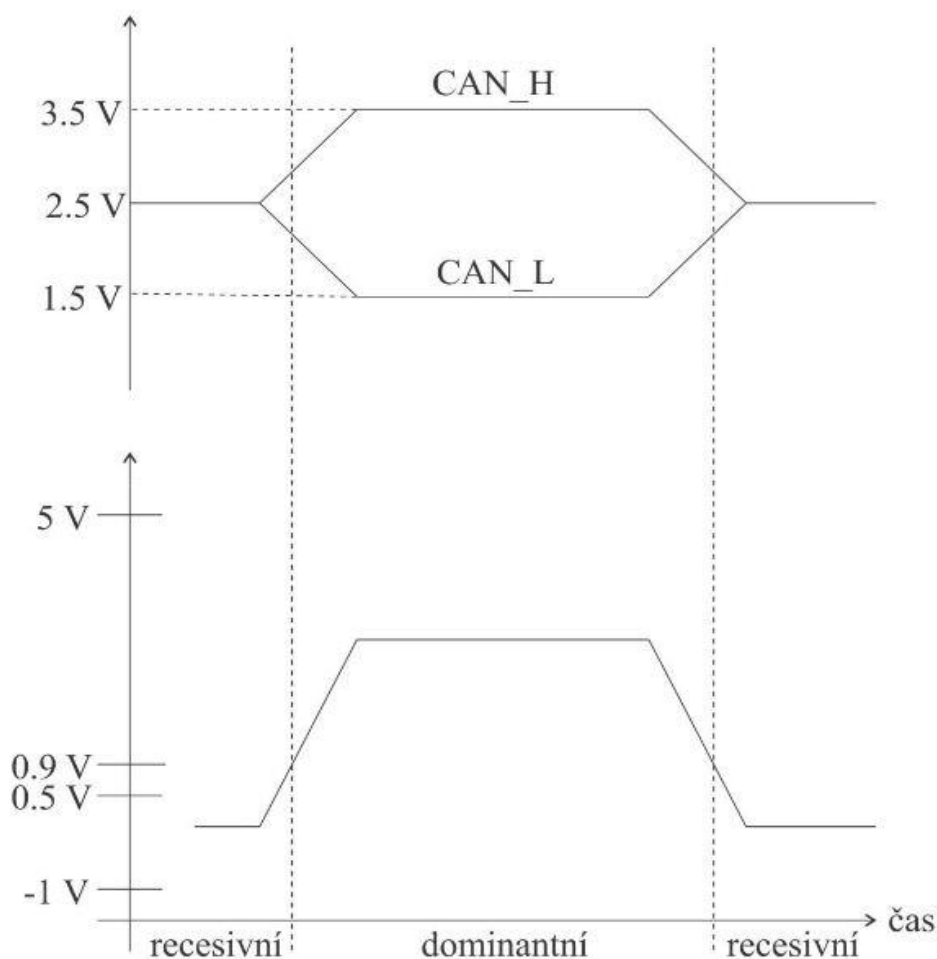


Obr. 1.2 Síť sběrnice CAN a její impedanční přizpůsobení

Pro uskutečnění takové sběrnice, se nejčastěji používá diferenciální sběrnice a ta je definovaná normou ISO 11898 a to platí včetně vysílacích budičů a přijímačů. Norma definuje princip časování, elektronické vlastnosti, kódování jednotlivých bitů a synchronizaci přenosu zpráv. Z předchozího obrázku je patrné, že se sběrnice sestavuje ze dvou vodičů, který se označují jako CAN_H a CAN_L. Na začátku a na konci vodičů se můžeme taky povšimnout rezistorů R1 a R2, které mají hodnotu 120 Ω . Tímto odporům se říká tzv. terminátory sběrnice CAN. Jejich funkce je zajišťovat impedanční přizpůsobení, a tím i vyrušení odrazů na vedení. [8]

1.3.3 Recesivní a dominantní stav

Jak už jsme si řekli sběrnice CAN může nabývat dvou stavů, recesivního (logická jednička) a dominantního (logická nula). Pokud máme všechny jednotky (uzly) propojené na sběrnici, to znamená že jsou v recesivním stavu, pak je i v tomto stavu sběrnice. Pokud se ale jedna jednotka (uzel) dostane do dominantního stavu, pak musí přejít do tohoto stavu i celá sběrnice. Jak víme sběrnice využívá diferenčního napětí mezi vodiči CAN_H a CAN_L. V recesivním stavu sběrnice je na obou vodičích stejné napětí. Při dominantním stavu je na sběrnici diferenční napětí o určité velikosti. Toto diferenční napětí chrání sběrnici proti elektrostatickým výbojům. Přenosová rychlost u rozhraní CAN není pevně daná, jelikož vysoce závisí na délce vedení. Ve vzdálenosti do 40 m se dá počítat i s rychlostí 1Mbit/s, ale pro vzdálenosti vyšší, než je 1000 m se rychlost pohybuje v řádech desítkách kbit/s a nižších. Na Obr. 1.3 je rozdíl mezi diferenčním napětím a napětím proti zemi. V dolní polovině obrázku je znázorněno právě diferenční napětí používané na sběrnici CAN. V horní polovině obrázku jsou znázorněna dvě napětí proti zemi. [8]



Obr. 1.3 Rozdíl mezi diferenčním napětím a napětím proti zemi [8]

1.4 Způsob řízení přestavníků

V současné době je přehazování přestavníků (výhybek) řešeno principem teplotní dilatace odporového drátu. Technologie umožňuje tichý a pomalý přehození přestavníku. Další technologií jsou relátka ty jsou poměrně rychlé. Poslední technologií jsou servomotory.

1.4.1 Odporový drát

Odporový drát funguje na principu materiálové roztažnosti. Když pustíme proud do odporového drátu, tak se nám začne zahřívat. Dojde k roztažnosti materiálu a přestavník se přehodí. Pokud chceme zpět přestavník musíme počkat až materiál zchladne a tím se vrátí přestavník do původní polohy.

Metoda je podle mě jenom experiment roztažnosti materiálů, nejde o spolehlivou metodu přehazování přestavníků.

1.4.2 Relé

Použití relé je lepší varianta, protože je rychlejší a spolehlivější než odporový drát. Nese s sebou spoustu nevýhod. První problém je, že nemůžeme nastavovat dorazy tzn. že nemůžeme volit libovolnou polohu přestavníků. Jako druhý problém je, že potřebujete nějaký mechanismus, co by po uvolnění relé vrátil zpět přestavník. Tento mechanismus by musel obsahovat nějakou pružinu která by jej vrátila.

Metoda je podle mě opět nevyhovující, protože obsahuje spoustu nevýhod pro práci přestavníků.

1.4.3 Servomotory

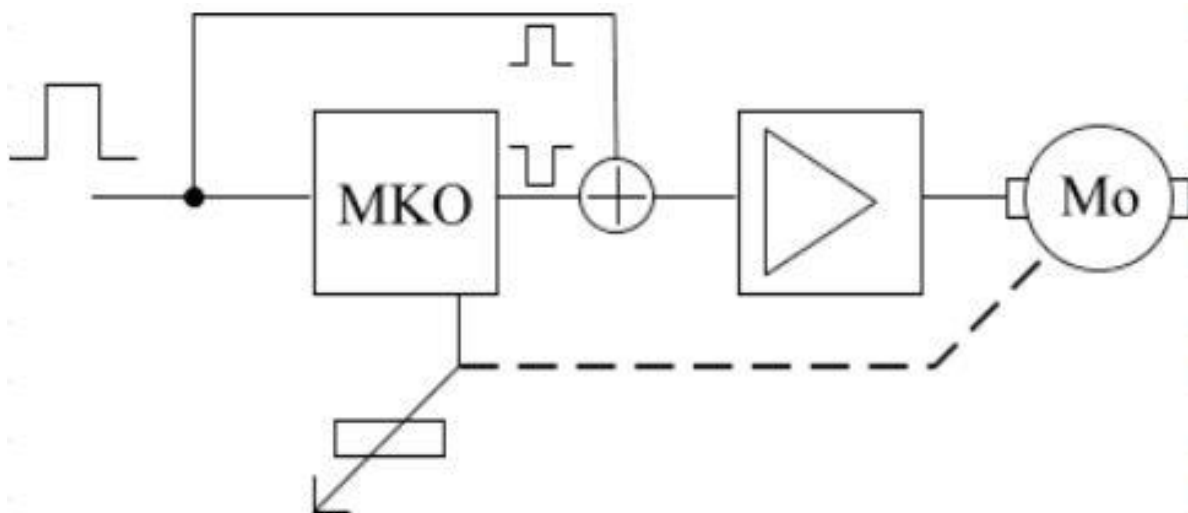
Servomotory jsou elektrické, hydraulické, pneumatické, a dokonce i parní. Rozdíl od běžných motorů je, že můžeme nastavit přesnou polohu natočení hřídele. Ovládají se jím například posuvy CNC strojů, čtecí hlavičky pevných disků a většina modelů RC používají servomotory.

Tato metoda je podle mě nejlepším řešením, protože si můžeme nastavovat libovolnou polohu přestavníků, a to bez nějakých komplikovaných mechanismů.

1.5 Servomotor

1.5.1 Blokové schéma servomotoru

Servomotory obsahují elektromotor s převodovkou a řídicí elektroniku. Jednoduché blokové schéma je na Obr. 1.4.



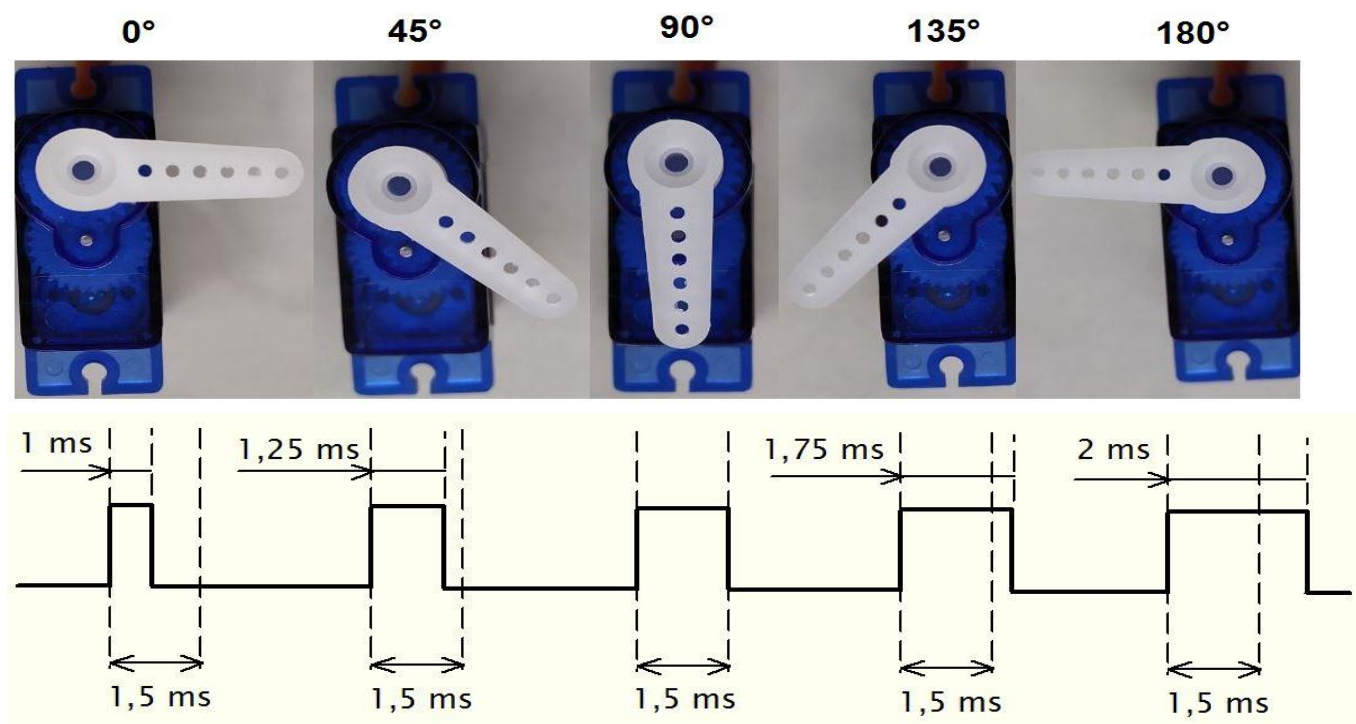
Obr. 1.4 Blokové schéma servomotoru [3]

Na vstup přichází řídicí impuls, který aktivuje monostabilní klopný obvod značeno na Obr. 1.4 jako MKO. Ten vygeneruje impuls o délce odpovídající aktuální poloze serva a opačné polarity, než je vstupní řídicí impuls. Tyhle dva impulsy se pak porovnávají a výsledkem bude rozdílový impuls. Ten je pak zesílen a přes můstkový spínač způsobí roztočení elektromotoru jedním nebo druhým směrem.

Elektromotor přes převodovku otáčí výstupní hřídeli a současně potenciometrem, který zde pracuje jako zpětná vazba do monostabilního klopného obvodu (MKO). Směr tohoto otáčení je takový, že impuls generovaný monostabilním klopným obvodem se svojí délkou přibližuje délce vstupního řídicího impulsu, dokud nejsou oba impulsy stejně dlouhé. Potom se elektromotor zastaví a servomotor dosáhl polohy, která odpovídá aktuálnímu přijímanému řídicímu impulsu.[3]

1.5.2 Reakce servomotorů na řídicí impuls

Řídicí impulsy pro servomotory se dnes pohybují o délce 1–2 ms. Z této myšlenky se dá lehce odvodit, že střední poloha servomotoru je 1,5 ms. Tedy jedna krajní poloha je 1 ms a druhá krajní poloha je 2 ms. Mechanické provedení servomotoru může být takové, že servomotor je schopen pohybu v rozsahu od 0° až do 180°. Zde chci zdůraznit, že to není pravidlem. Servomotory mohou mít různý rozsah pohybu. Většina servomotorů má taky takzvané mechanické blokování. Mechanické blokování je na koncích servomotorů. Zde musíme být opatrní, protože přetáčení serva může způsobit zvýšení odebíraného proudu a může pak dojít k zničení servomotoru. Na Obr. 1.5 je ukázka natáčení serva na řídicí impulsy. [3]

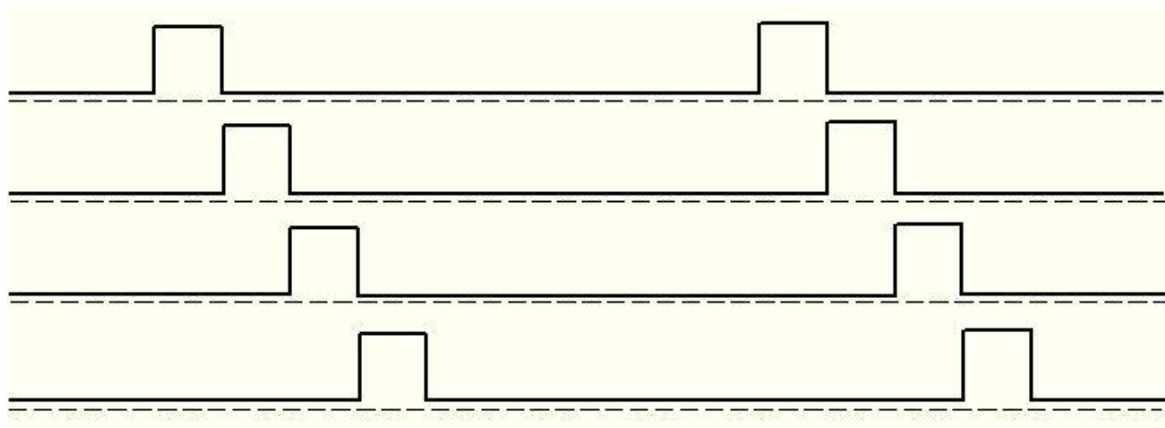


Obr. 1.5 Natáčení servomotorů řídicími impulsy

1.5.3 Řízení servomotorů

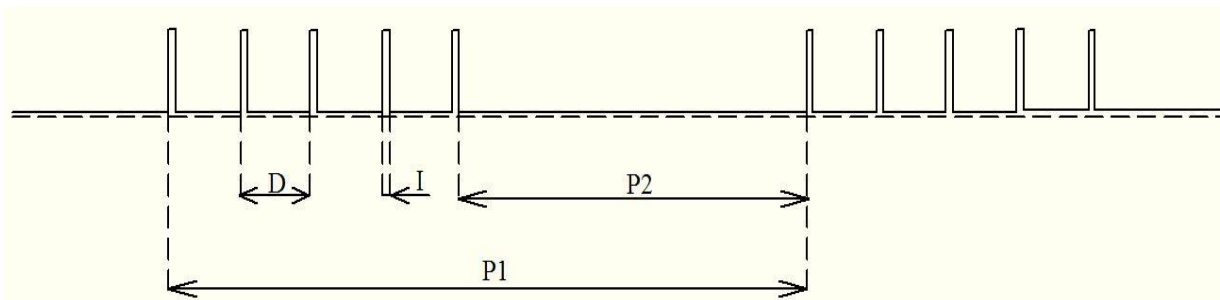
Servomotor má tři přívodní kabely, a to většinou černý, červený a žlutý. Na konci je konektor, který má rozteč dutinek 2,54 mm. Černý vodič slouží jako GND (nebo hnědý vodič). Červený vodič je napájení 4,8V – 5 V. Žlutý vodič je vstup pro řídicí impuls. Typů servomotorů je podstatně více.

Servomotory pracují s kladnými řídicími impulsy od délce 1–2 ms. Impulsy se opakují s frekvencí 50 Hz. Pokud budeme posílat méně řídicích impulsů, tak zpomalíme rychlost pohybu serva. Velikost impulsů je rovna velikosti napájecího napětí serva. Na Obr. 1.6 jsou zobrazeny impulsy pro čtyři serva, která se opakují přibližně v rytmu 50 Hz.



Obr. 1.6 Řídicí impulsy pro čtyři servomotory

Na předchozím obrázku jsou zobrazeny čtyři průběhy (pro čtyři serva). Tyto průběhy se zpracovávají v kodéru a jeho výstupní signál můžeme vidět na Obr. 1.7

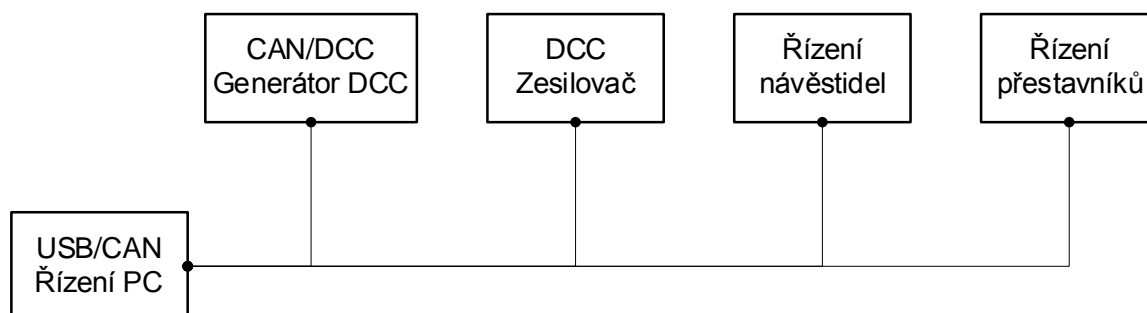


Obr. 1.7 Výstup kodéru pro čtyři servomotory

Podle typu kodéru zůstává konstantní buď P1 a skupiny impulsů se opakují v pravidelném rytmu kolem 50 Hz, nebo je konstantní mezera mezi skupinami impulsů P2, ta se pohybuje většinou kolem 10–15 ms. Označení D je délka řídicího impulsu pro jedno servo to je přibližně 1–2 ms. V obrázku je oddělovací impuls přibližně 0,25 – 0,3 ms dlouhý a je označený písmenem I. [3]

2 Koncepce a blokové schéma řízení kolejiště

Pro řízení kolejiště je vymyšlena struktura na blokovém schématu viz. Obr. 2.1



Obr. 2.1 Blokové schéma kolejiště

2.1 Blok USB/CAN

Řízení kolejiště při běžném provozu zajišťuje program běžící na PC. Program má srozumitelné uživatelské prostředí, to složí k sledování stavu kolejiště a jeho funkčních jednotek. Můžeme manuálně ovládat prvky kolejiště a také nastavovat automatické spouštění potřebných sekvencí. [7]

2.2 Blok CAN/DCC

Generátor signálu DCC přijímá příkazy určené do kolejí v podobě CAN zpráv, z těchto zpráv vytváří signál DCC. Ke každému příkazu přidává nadbytečnost, příkazy následně ukládá a neustále je opakuje. Jednotka po rozhraní CAN odesílá zprávy v podobě watchdogu k centrálnímu řízení. Centrální řízení při výpadku periferie vypne v bloku DCC Repeater což je buzení kolejí, jinak by se vlaky rozjely na stejnosměrné napětí. Časovač této periferie také hlídá, jestli neustále přichází CAN zprávy pro DCC. Pokud by došlo k výpadku, tak nastaví rychlosti všech vlaků v paměti na 0. [7]

2.3 Blok DCC Zesilovač

Blokem DCC zesilovač je množství jednotek, z nichž každá využívá až 8 úseků kolejí signálem DCC. Tento signál přijímají jednotky z generátoru DCC a zesilují ho pro napájení lokomotiv. Dále se sleduje odběr proudu na každém úseku, to slouží pro detekci obsazenosti úseků. Hodnoty pak posíláme do centrálního řízení pro rozhraní CAN. Buzení kolejiště zajišťuje step-down měnič napětí s H-můstkem, který řízený přímo signálem DCC.

Pomocí sériově zařazeného odporu, zesilovače a AD převodníku s odpovídající rychlostí a přesností se realizuje měření proudu. S tímto systémem jsme schopni detekovat odběr již od řádu mA. [7]

2.4 Blok řízení návěstidel

V tomto bloku se jedná o řízení návěstidel. Každá z jednotek přijímá příkazy z CAN rozhraní, tím budí osvětlení a návěstidla na vymezené části kolejiště. Tato jednotka byla sdružena s jednotkou řízení přestavníků. [7]

2.5 Blok řízení přestavníků

Z CAN rozhraní přijímají příkazy budiče přestavníků, podle kterých se přehazují přestavníky. V této práci je projektován přestavnickový kontrolér s využitím servopohonů pro spolehlivější přehození přestavníků. K tomuhle bloku spadá i řídicí jednotka točny. [7]

3 Praktická realizace

V této části budou popsány důvody volby jednotlivých komponent a způsoby výsledné realizace.

3.1 Požadavky na řídicí jednotku přestavníků

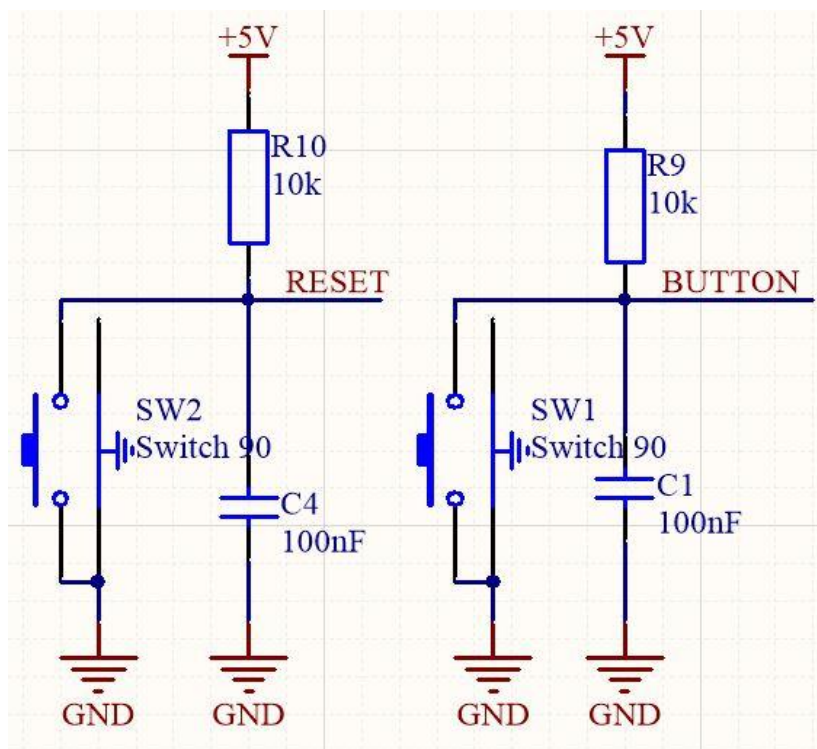
Od řídicí jednotky přestavníků požadujeme řízení servomotorů. Dále resetování jednotky, nastavení času mezi otáčením servomotoru, zpětnou vazbu na vyžádání, nastavení dorazů pro servomotory, výběr a přehození výhybek a nastavení čísla jednotky. Všechny tyto funkce jsou popsány detailně v kapitole 0.

3.2 Návrh zapojení řídicí jednotky přestavníků

Návrh zapojení pro řídicí jednotku přestavníku byl navržen v počítačovém programu Altium Designer 18. Navržený vzorového projektu vychází z internetových stránek Projekty FEL. V projektu je použita standardizovaná deska DPS, zejména umístění konstrukčních rozměrů a rozměry desky. Standardizován je taktéž konektor RJ-45 pro propojení DPS pomocí UTP kabelu s nadřazeným systémem. Protože konektor RJ-45 není již použitý, tak bylo zapojení konektoru RJ-45 pozměněno, podle již zakoupeného a do budoucna plánovaného konektoru. Jeden port slouží pro připojení následující jednotky a druhý port konektoru RJ-45 slouží pro připojení předchozí jednotky. V návrhu desky se nachází header 2x4 s roztečí 2,54 mm s pevně danou pozicí na desce plošných spojů. Funkce tohoto headeru spočívá v tom, že můžeme vrstvit jednotky na sebe i pod sebe a není nutno je propojovat UTP kabelem. Využíváme to u jednotek, které se nachází v blízkosti vedle sebe. U konektoru RJ-45 jsou umístěny dva napájecí konektory a ty jsou určeny k napájení desky plošných spojů, kdyby byl UTP kabel proudově přetěžován. U napájecích konektorů je zelená LED, která slouží k indikaci jednotky pod napětím.

3.2.1 Tlačítko

Pro jednotku jsem navrhnul dvě tlačítka. Jedno tlačítko funguje jako RESET mikrokontroléru MC9S08DZ60. Druhé tlačítko pro manuální ovládání jednotky BUTTON. Na Obr. 3.1 je zapojení tlačítek.

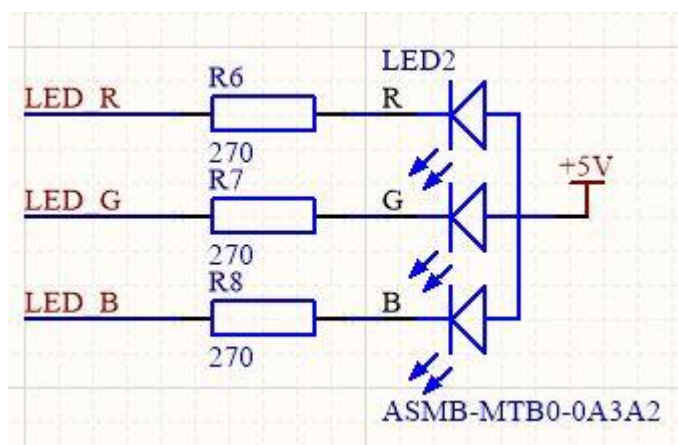


Obr. 3.1 Schéma zapojení tlačítek

V zapojení jsou rezistory R9 a R10 s hodnotou 10k Ω . Kondenzátory C1 a C4 jsou vyhlazovací kondenzátory napět'ového zákmitu.

3.2.2 RGB LED

RGB LED je navržnuta pro indikaci stavů na řídicí jednotce přestavníků. Její zapojení je velmi jednoduché a je na Obr. 3.2.

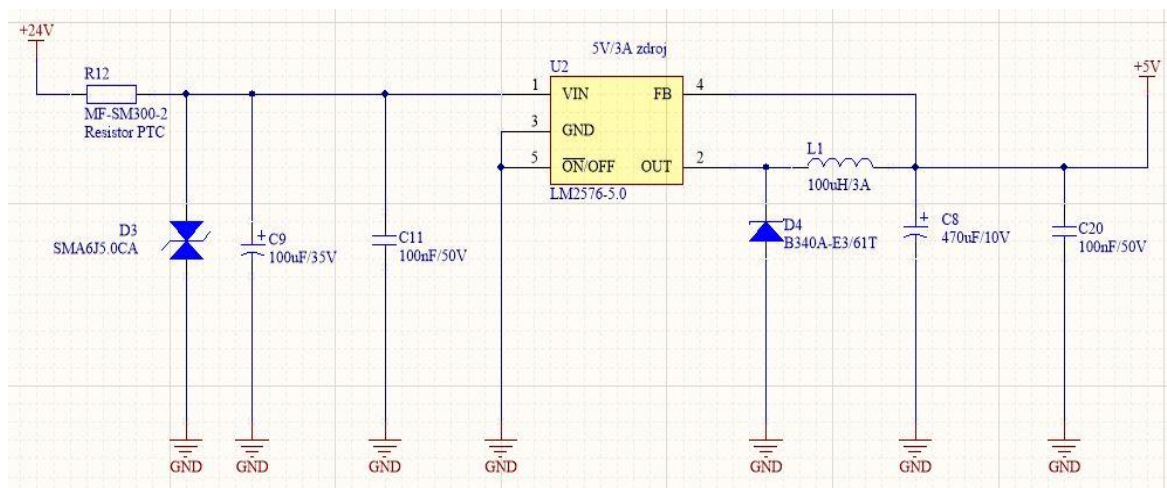


Obr. 3.2 Schéma zapojení RGB LED

V zapojení jsou odpory R6, R7 a R8 o hodnotě 270 Ω . RGB LED má tu výhodu, že můžeme svítit až šesti barvami na jedné LED. Kombinací barev červené, zelené a modré jsme schopni zobrazit další barvy, a to jsou azurová, bílá, fialová a žlutá.

3.2.3 Zdroj napětí

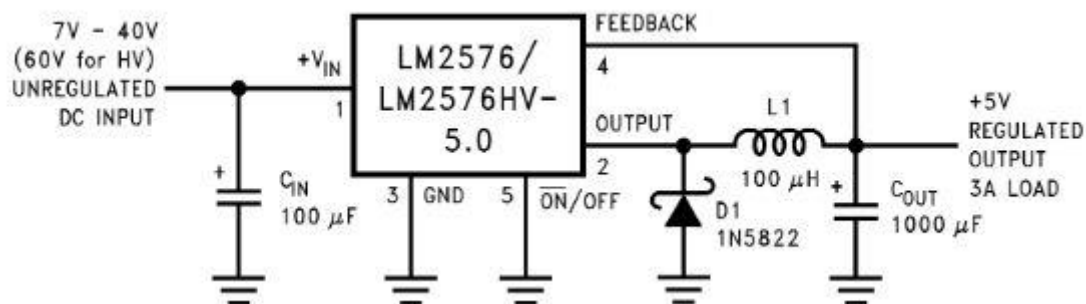
Na desku řídicí jednotky přestavníku je přivedeno z modelové železnice 24 V. Toto napětí musíme snížit na vhodnou napěťovou úroveň. Používám mikrokontrolér MC9S08DZ60, ten požaduje napájecí napětí 5 V. Servomotory jsou napájeny taktéž napájecím napětím 5 V. Tlačítka, RGB LED a další komponenty jsou napájeny napájecím napětím 5 V. Proto nám bude stačit jeden snižující spínaný regulátor napětí (spínaný step down regulátor) s výstupním napětím 5 V a výstupním proudem 3 A. Zvolil jsem regulátor LM2576S-5.0 který má účinnost 77 %. Maximální vstupní napětí je 45 V. Na Obr. 3.3 můžeme vidět zapojení spínaného step down regulátoru.



Obr. 3.3 Schéma zapojení spínaného step down regulátoru

Zapojení regulátoru je navrženo z veliké části podle doporučeného zapojení v katalogovém listu (datasheet) [4], v kterém je doporučeno na vstup (1) připojit elektrolytický kondenzátor C9 o hodnotě 100µF. Slouží k vyhlazení vstupního napětí. Kondenzátor musí splňovat ještě jednu hodnotu a tu, že vydrží napětí 24 V, proto je zde zvolen kondenzátor o hodnotě 100µF/35V. Dále se uvádí, že musíte uzemnit piny integrovaného obvodu GND (3) a ON/OFF (5). Pin OUT (2) je výstupní část. Na tu je doporučeno připojit Schottkyho diodu D4, protože má krátkou dobu zotavení a cívku L1 o hodnotě 100µH. Za cívku se připojí feedback integrovaného obvodu (4) a připojením kondenzátoru C8 o hodnotě 1000µF výrobce dále už nic neuvádí. Dioda D4 slouží k uzavření obvodu, protože při rozepnutí spínaného zdroje se otočí polarita napětí na cívce. Dioda D4 se stane propustnou a začne se akumulovat energie na kondenzátoru C8 a ta se následně spotřebuje na zátěži. Při sepnutí spínače (tranzistor) mezi vstupem a výstupem je dioda D4

v závěrném směru, takže nic nepropustí a akumuluje se na cívce energie. Zdroj je schopný dodávat 3 A. Na Obr. 3.4 je doporučené zapojení.

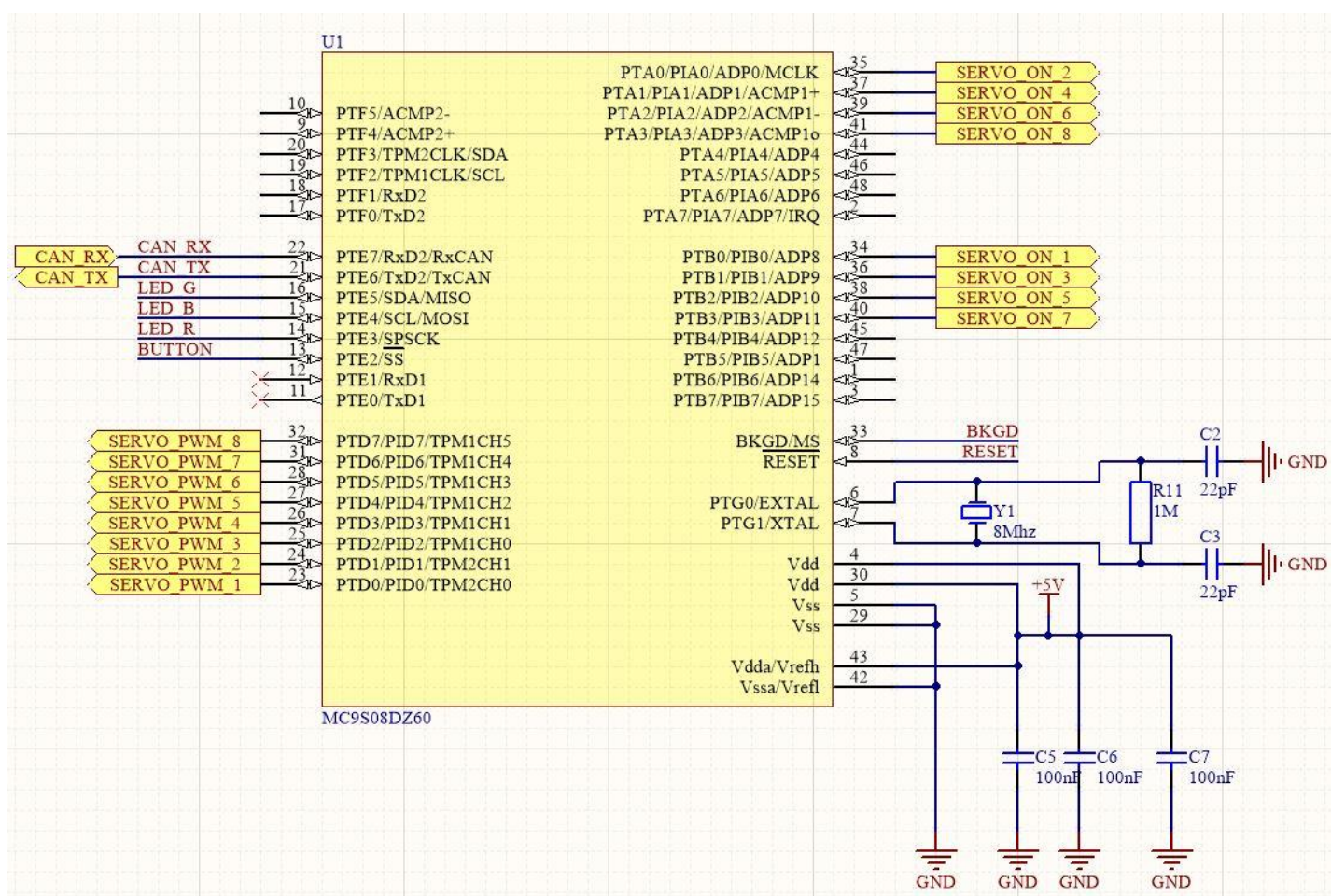


Obr. 3.4 Schéma doporučeného zapojení [4]

Toto zapojení jsem rozšířil o kondenzátor C11 a C20 o hodnotě 100nF/50V. Tento kondenzátor funguje jako blokovací kondenzátor a způsobí zvýšení stability regulátoru tzn. že vyhladí vysoké napěťové zákmity. Dále jsem zvolil jinou hodnotu filtračního kondenzátoru C8, která je 470µF/10V. Kondenzátor je prostorově menší a funkci to nijak nemění. Dále jsem přidal transil D3, který zajišťuje ochranu před přepětím na vstupu zdroje. Poslední součástíkou je zde PTC termistor R12, který plní funkci vratné pojistky.

3.2.4 Mikrokontrolér

Jednou z nejdůležitějších částí je zapojení mikrokontroléru. Pro řídicí jednotku přestavníků jsem zvolil mikrokontrolér MC9S08DZ60 od výrobce NXP Semiconductors. Tento mikrokontrolér má 16 ADC kanálů, 48 pinů, programovací paměť o velikosti 60 kB, RAM s pamětí 8 kB, osmibitovou datovou sběrnici, 8 kanálů TPM, zabudovaný COP watchdog a MSCAN a další periferie. Zapojení mikrokontroléru je na Obr. 3.5.



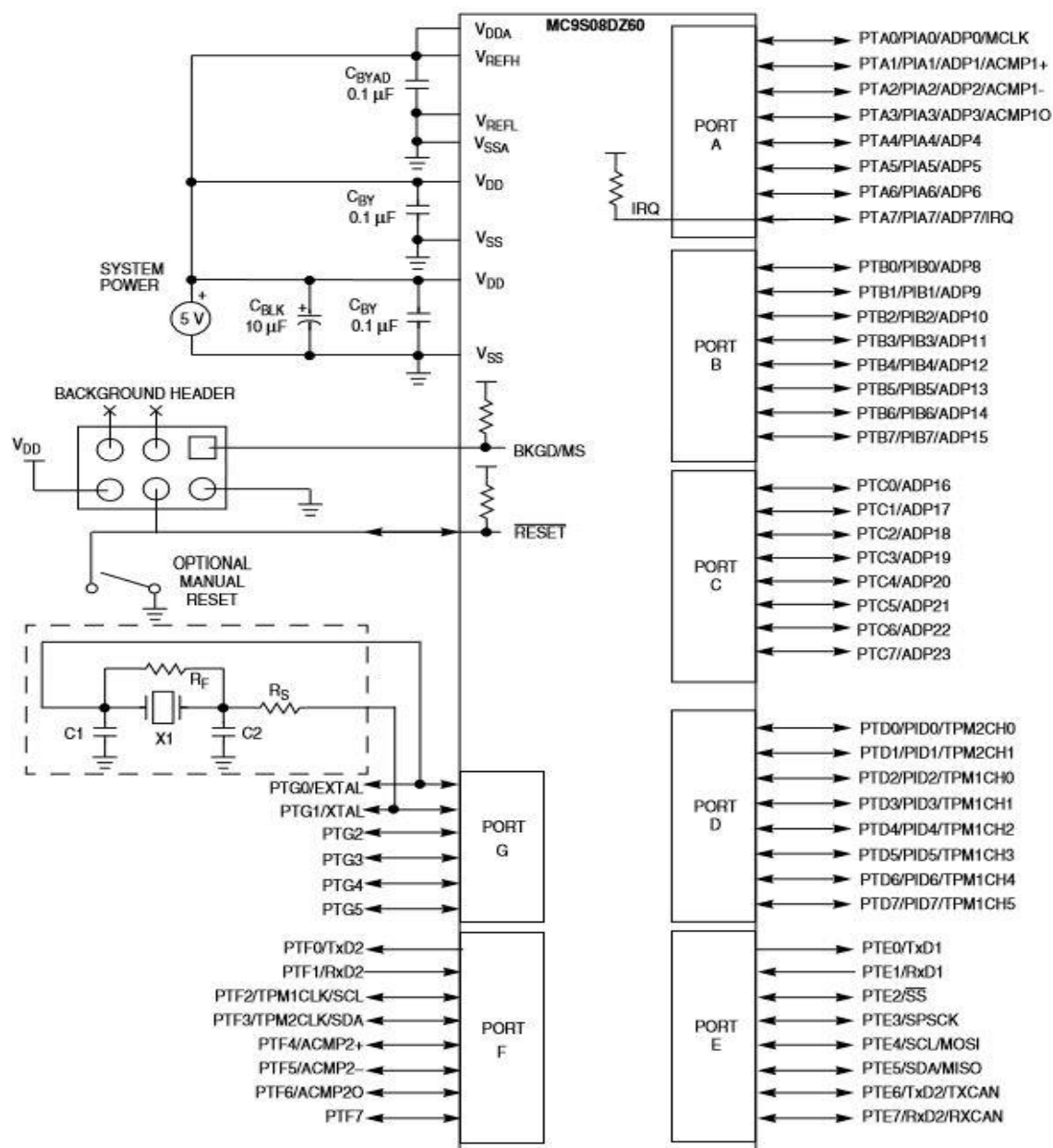
Obr. 3.5 Schéma zapojení mikrokontroléru

Navrhoval jsem podle doporučeného zapojení, které přikládá výrobce. Na Obr. 3.6 je doporučené zapojení od výrobce [6].

Na vstupní piny PTG0/EXTAL a PTG1/XTAL (PORT G) připojíme krystalový oscilátor, který označený jako Y1 s frekvencí 8 MHz. Dále musíme připojit odpor R11 o hodnotě $1\text{M}\Omega$ a k tomu kondenzátory C2 a C3 o hodnotě 22pF . Je to z důvodů, že kondenzátory se chovají jako zatěžovací kapacita a vytvoří se tak rezonanční obvod. Hodnota kondenzátoru je specifikována od výrobce krystalů. Aby byla požadovaná frekvence přesná, obvod oscilátoru musí mít stejnou zatěžovací kapacitu, na kterou je krystal nastaven.

Zapojení pro řídicí jednotku přestavníku je následující. Nejprve musíme zapojit komunikaci pro sběrnici CAN. K tomu slouží piny Rx_CAN (vstupní) a Tx_CAN (výstupní). Piny PTE5, PTE4 a PTE3 slouží k řízení RGB LED. Pin PTE2 je využit pro

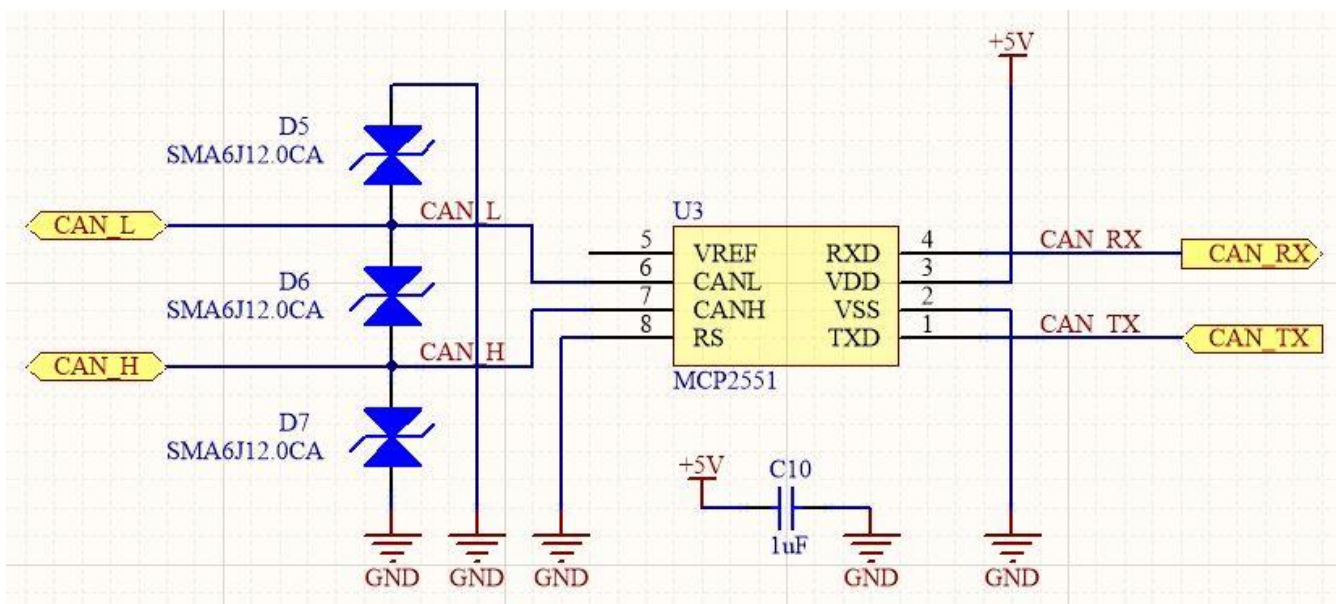
tlačítko BUTTON. Jak už jsem zmiňoval v předchozích řádcích, tak mikrokontrolér má k dispozici osm PWM kanálů, které můžeme vidět taky na Obr. 3.6, jedná se o PORT D. Toho využijeme pro generování signálu pro servomotory, kterých je na řídicí jednotce taktéž osm. Port D je plně využitý a jedná se o výstupní porty pro řízení servomotorů (nastavujeme hodnotu natočení). Poslední krok v zapojení je dát pokyn ke spuštění servomotoru. To uděláme tak, že využijeme porty například z řady PORT A a PORT B. Přiřadíme jim funkci výstupní. Je to z důvodů zapínání nebo vypínání servomotorů. Pin BKGD složí k nahrání programu z programátoru a pin RESET slouží k resetování mikrokontroléru.



Obr. 3.6 Schéma zapojení mikrokontroléru doporučené výrobcem [6]

3.2.5 Obousměrný budič CAN sběrnice

Budič převádí signály, které vygeneroval mikrokontrolér na signály, které jsou vhodné pro komunikaci po sběrnici. Taky chrání mikrokontrolér před napěťovými špičkami vzniklými na sběrnici a rušením. Budič umožňuje komunikaci po sběrnici o rychlosti 1 Mb/s. Funguje taky jako přijímač i vysílač. Pro obousměrný budič sběrnice jsem zvolil MCP2551. Na Obr. 3.7 je schéma zapojení obousměrného budiče. Pro zapojení budiče jsem opět využil doporučené zapojení výrobcem. Zapojení je v Tab. 3.1.



Obr. 3.7 Schéma zapojení obousměrného budiče

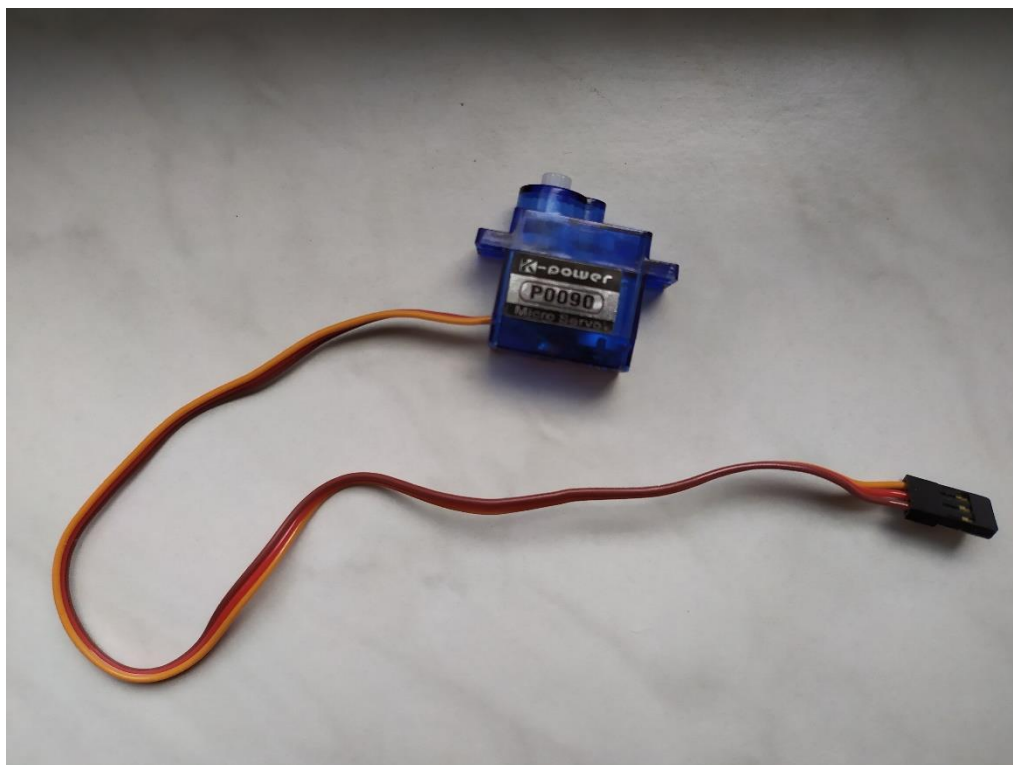
Tab. 3.1 Zapojení MCP2551 podle výrobce [5]

Číslo pinu	Název pinu	Funkce pinu
1	TXD	Vstup pro CAN_TX
2	VSS	Uzemnění (GND)
3	VDD	Napájecí napětí (5 V)
4	RXD	Výstup pro CAN_RX
5	VREF	Reference (posudek) výstupního napětí
6	CANL	Úroveň napětí CANL
7	CANH	Úroveň napětí CANH
8	RS	Kontrola vstupních hran

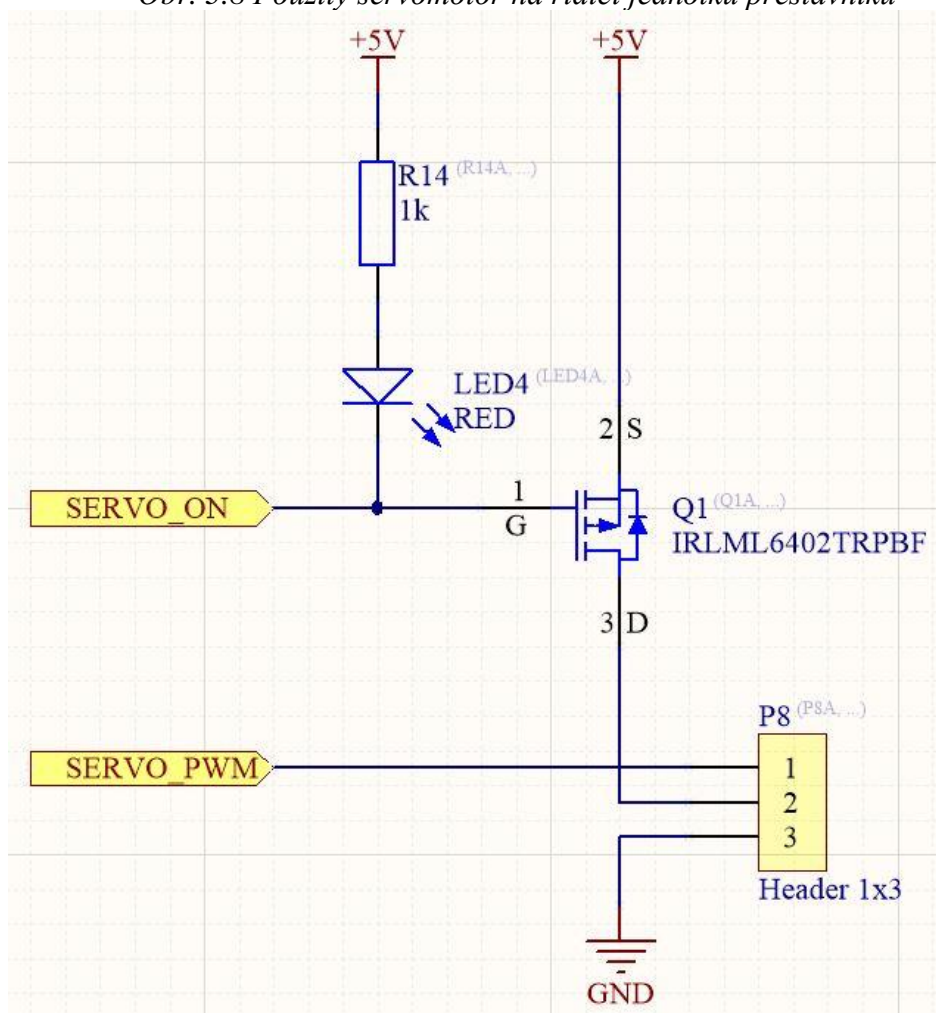
Z tohoto důvodu jsem zapojil VSS (2) a RS (8) na zem (GND). Na VDD (3) jsem přivedl napájecí napětí 5 V, kde je připojený kondenzátor C10 o hodnotě 10 μ F, který slouží jako vyhlazovací kondenzátor. Na TXD (1) jsem připojil CAN_TX a na RXD (4) jsem připojil CAN_RX. VREF jsem nezapojil, protože nepotřebuji referenci výstupního napětí. Mezi CAN_L (6) a CAN_H (7) jsou zapojeny transil diody D5, D6 a D7, které fungují jako přepěťové ochrany. Na sběrnici CAN se nachází dva stavy, dominantní a recesivní. Pokud je rozdílové napětí nižší, jak definované napětí mluvíme o recesivním stavu, hodnota stavu je 0 V. Pokud je rozdílové napětí mezi CAN_L a CAN_H větší jak definované napětí, pak mluvíme o dominantním stavu, hodnota stavu je jiná než 0 V. Recesivní a dominantní stavy odpovídají příslušným logickým stavům. Recesivní stav chápeme jako logickou jedničku na vstupním pinu TXD (1). Pokud je na vstupním pinu TXD (1) logická nula, pak je budičem uvedena sběrnice do dominantního stavu. RXD (4) pořád sleduje rozdíl mezi CAN_L a CAN_H, jestliže je sběrnice v dominantním stavu, pak je na výstupním pinu RXD (4) logická jednička. RS (8) umožňuje výběr mezi třemi režimy budiče. První je režim Standby mode, který lze vybrat připojením logické jedničky na pin RS. Budič pouze přijímá. Druhým režimem je Slope-Control. Redukuje elektromagnetické rušení pomocí náběžných a sestupných hran CAN_L a CAN_H. Připojením rezistoru mezi pin RS (8) a zem zvolíme druhý režim. Třetí režim je High-Speed. Pokud je pin RS (8) připojený k pinu VSS (2), pak tento režim umožňuje maximální přenosovou rychlost 1 Mb/s.

3.2.6 Servomotory

Poslední klíčovým zapojením jsou servomotory. Na Obr. 3.8 je použitý servomotor při práci na modelové železnici pro přehazování přestavníku (výhybek) a Obr. 3.9 je schéma zapojení ovládání servomotoru. Pro lepší pochopení rozdělím pomyslně schéma na dvě části. První část bude pro konektor P8 (řídicí část obvodu). To je konektor kam se připojí servomotor a druhá část je pro tranzistor MOSFET, značeno ve schématu jako Q1 (spouštěcí část obvodu).



Obr. 3.8 Použitý servomotor na řídicí jednotku přestavníků



Obr. 3.9 Schéma zapojení ovládání servomotoru

Řídicí část obvodu je pro konektor P8 (Header 1x3) má tři vývody a slouží jako výstup pro servomotory. Jeden pin (3) je uzemnění (GND). Na tenhle pin přivedeme hnědý vodič (GND). Druhý pin (2) připojíme na tranzistor MOSFET (3) tento výstup u tranzistoru nazýváme DRAIN. Tento výstup napájí servomotor 5 V a připojíme na něj červený vodič servomotoru. Třetí pin (1) konektoru P8 připojíme na žlutý vodič servomotoru. Na tento pin (1) přivádíme výstup SERVO PWM signálu z mikrokontroléru.

Spouštěcí část obvodu je pro tranzistor MOSFET. Na pin (2) SOURCE se přivede napájecí napětí 5 V. Pin (3) jsme řešili v předchozích řádcích, pro upřesnění se zapojuje na konektor P8. Pin (1) GATE je řídicí část (zapnuto nebo vypnuto). Připojil jsem zde červenou LED4 pro indikaci. Při běhu motoru tato LED svítí. Musíme připojit k LED také předřadný odpor R14 o hodnotě $1k\Omega$, aby nedošlo ke zničení LED. Na pin (1) GATE je přiveden výstup z mikrokontroléru SERVO ON.

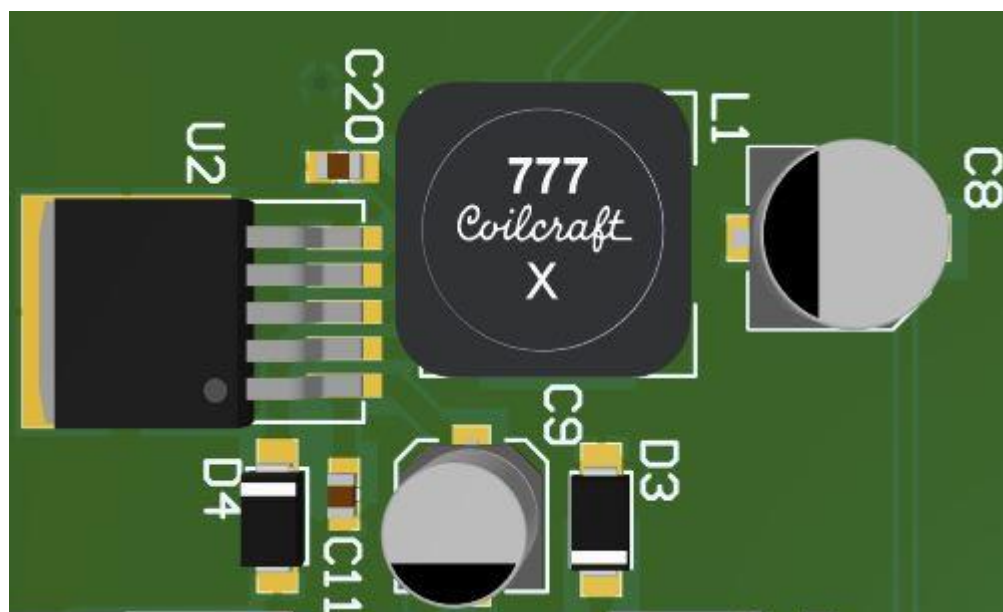
Když přivedeme na SERVO ON logickou jedničku, pak nesvítí LED4 a nedojde k sepnutí tranzistoru Q1, tudíž je servomotor vypnutý. Pokud přivedeme logickou nulu, pak svítí LED4, zároveň se sepne tranzistor Q1 a přivede na konektor P8 (2) napětí 5 V pro servomotor. Tím je servomotor v zapnutém stavu.

3.3 Návrh DPS

Deska plošných spojů je součástí snad všech elektronických zařízení. Základní funkce je vytvořit vodivé propojení mezi vývody jednotlivých součástek. Při návrhu DPS je si třeba uvědomit jaká bude použita technologie pájení a jaké bude umístění součástek na desce. Technologie dělím na tři. První technologie je ruční pájení. Druhá technologie je pájení pomocí vlny. Třetí technologie je pájení v reflow peci. Součástky dělím podle pouzdra, a to na dva typy. Na pouzdra s vývody (např. THT) a bez vývodů (např. SMD).

3.4 Postup návrhu

Jako první krok jsem musel zjistit, jestli se mi na desku plošných spojů vejdu součástky a jestli bude dostatečný prostor pro propojování mezi součástkami. Druhým krokem jsem si nastavil pravidla pro routing. Potom jsem umístil mikrokontrolér a konektor RJ-45. Zvolil jsem výhodnou pozici zdroje na DPS a měl jsem připraveno. Teď následovalo rozmístění filtračních a blokovacích kondenzátorů. Poté jsem zvolil výhodnou pozici pro servomotory, bylo třeba zajistit, aby to bylo na kraji desky. Důvod je ten, aby byla lehká dostupnost k výstupním konektorům servomotorů. Potom následovalo rozmístění zbylých součástek a jejich propojení. Na Obr. 3.10 je ukázka spínaného zdroje LM2576S-5.0 z DPS.



Obr. 3.10 Ukázka DPS spínaný zdroj LM2576S-5.0

Na Obr. 3.10 můžete vidět, jak jsem rozmístil blokovací kondenzátory C20 a C11. Jsou co nejbliže k spínanému zdroji LM2576S-5.0, aby odfiltrovaly vysokofrekvenční šum. Kondenzátory C8 a C9 jsou elektrolytické. Dioda D3 je transil a dioda D4 je Schottkyho dioda. L1 je cívka o hodnotě 100 μ H.

4 Komunikační protokol

Jednou z nejdůležitějších částí této práce je právě komunikační protokol. Komunikační protokol slouží jako návod k použití řídicí jednotky přestavníků. Díky těmto řádkům může docházet k vylepšením do budoucna a věřím, že budoucí kolegové budou právě na tomto komunikačním protokolu stavět svoje nápady, myšlenky a postupně dosáhnou vrcholového díla.

V této kapitole je popsána komunikace řídicí jednotky přestavníků po sběrnici CAN s nadřazeným systémem. Dále jsou popsány funkce, které se očekávají od jednotky přestavníků.

4.1 Komunikace přes CAN

4.1.1 Zapojení CAN

Komunikační rychlost obousměrného budiče pro sběrnici CAN je nastavena na 500 kbit/s. Obousměrný budič sběrnice CAN je zapojen tak, aby byl zvolen v režimu High Speed, který je popsán v kapitole 3.2.5.

4.1.2 Vytvoření CAN ID

Komunikační systém na modelové železnici využívá přiřazení ID jednotlivým zprávám s ohledem na to, která jednotka je poslala a komu jsou přiřazeny. ID má velikost jedenácti bitů. Toto ID je rozděleno na dvě části. První čtyři bity ID slouží pro určení typu zprávy. Tyto zprávy posílá systém řídicí jednotce a označují se písmenem W (z anglického slova write, který v překladu znamená napsat). V případě, kdy je z řídicí jednotky zpráva vyslána a je určena nadřazenému systému, pak se označuje písmenem R (z anglického slova read, který v překladu znamená číst). V tomto případě můžeme použít až šestnáct jednotek. V Tab. 4.1 je ukázka rozdělení typů zpráv mezi zatím použitými řídicími jednotkami modelové železnice. V Tab. 4.1 můžeme vidět typy jednotek, který mají přiřazenou ID a taky jaký směr vysílání mají k dispozici. Pod zprávu označenou číslem 0 je Nouzový stav. Nouzový stav má jenom zapisovací funkci (W), jeho funkcí je okamžitě zastavit veškerý provoz na celé železnici, kdyby došlo k nějakému kolapsu. Další zprávy 1 až 11 jsou zprávy vyhrazené pro jednotlivé jednotky. Pro jednotku přestavníků (výhybek) je vyhrazena zpráva 7 a 8 je zde k dispozici zápis do jednotky (W) i výpis z jednotky (R). Poslední typy zpráv jsou označené

čísla 12 až 15, ale ty jsou zatím nevyužity a jsou tedy připraveny na další rozšíření modelového kolejiště.

Tab. 4.1 Přiřazení adres konkrétním typům řídicích jednotek

První čtyři bity [Dec]	Celé ID [Hex]	Typ jednotky	Směr vysílání dat
0	0x01	Nouzový stav	Zápis do jednotky (W)
1	0x81	DCC Generátor	Zápis do jednotky (W)
2	0x101		Výpis z jednotky (R)
3	0x181	DCC Zesilovač	Výpis z jednotky (R)
4	0x201		Výpis z jednotky (R)
5	0x281		Zápis do jednotky (W)
6	0x301		Zápis do jednotky (W)
7	0x381	Kontrolér výhybek	Zápis do jednotky (W)
8	0x401		Výpis z jednotky (R)
9	0x481	Kontrolér návěstidel	Zápis do jednotky (W)
10	0x501	Kontrolér točny	Zápis do jednotky (W)
11	0x581		Výpis z jednotky (R)
12	Nevyužito	Nevyužito	Nevyužito
13			
14			
15			

Jak už jsem zmiňoval v předchozích řádcích, tak ID je jedenácti bitové. První čtyři bity jsou pro rozpoznání jednotky a zbylých sedm bitů je vyhrazeno pro přesnou identifikaci řídicí jednotky tzn., že máme k dispozici 128 jednotek stejného typu. Pro upřesnění máte binárně sedm jedniček 000 0111 1111 a to udává dekadické číslo 128. Nadřazený systém posílá tři typy zpráv. První typ zprávy je nouzové vysílání, které bylo zmíněno v předchozích řádcích. Druhý typ zprávy je zasílat zprávy jednotkám stejného typu. Třetí typ zprávy, je zpráva určená konkrétní řídicí jednotce. Pro snadnější řízení je využívána hexadecimální podoba ID (Celé ID [Hex]). Každá zpráva má tedy určitou adresu v hexadecimálním čísle. Pro zjištění, pod jakým ID číslem bude řídicí jednotka zapisovat zprávy nebo pod jakým ID bude vypisovat, stačí k přiřazeným ID přičíst příslušné číslo jednotky v hexadecimálním tvaru. ID jsou přiřazeny k jednotkám s ID číslem 1. Pro zprávu 7 se napíše jedenácti místné binární číslo, které je 011 1000 0001 a převod do hexadecimálního tvaru je 0x381.

4.1.2.1 Vytvoření ID pro řídicí jednotku přestavníků

Pro řídicí jednotku přestavníků (výhybek) máme v Tab. 4.1 důležité informace. Tyto informace jsou vypsány v Tab. 4.2. Příklad vytvoření CAN ID pro řídicí jednotku přestavníků je v Tab. 4.3.

Tab. 4.2 Přiřazení adres pro řídicí jednotku přestavníků

První čtyři bity [Dec]	Celé ID [Hex]	Typ jednotky	Směr vysílání dat
7	0x380	Výhybkový kontrolér	Zápis do jednotky
8	0x400		Výpis z jednotky

Tab. 4.3 Vytvoření CAN ID

Typ zprávy				Číslo jednotky							
10	9	8	7	6	5	4	3	2	1	0	11 bitů
1	0	0	0	0	0	0	0	1	1	0	0x406

Řekněme, že chceme řídicí jednotku přestavníků číslo 6 (šestá řídicí jednotka) a směr vysílání dat bude R (výpis z jednotky). V Tab. 4.4 jsou shrnuty naše požadavky pro práci k vytvoření CAN ID v Tab. 4.3.

Tab. 4.4 Příklad požadavků

Požadavek	Parametr
Typ jednotky	Výhybkový kontrolér
Typ zprávy (první čtyři bity) [Dec]	8
Číslo jednotky	6
Směr vysílání dat	Výpis z jednotky

V Tab. 4.3 do typu zprávy (**Typ zprávy** pouze první čtyři bity) napíšeme číslo 8, ale v binární soustavě tedy 1000. Do čísla jednotky (**Číslo jednotky** dalších sedm bitů) napíšeme číslo 6, ale opět v binární soustavě tedy 000 0110. Naše výsledné CAN ID v binární podobě jedenácti bitů je 100 0000 0110 a převedením tohoto binárního čísla do hexadecimální soustavy dostaneme číslo 0x406.

4.1.2.2 Přiřazení CAN ID jednotce pro určitý typ dat

Každý typ zprávy má přesně danou délku bytu v Tab. 4.5 je napsána délka bytu pro jednotlivé ID zprávy.

Tab. 4.5 Přiřazení ID jednotlivým typům zpráv

ID [Hex]	Délka [byte]	Směr vysílání dat	Význam dat
0x386	8	Zápis do jednotky	Nastavení výhybkového kontroléru
0x406	8	Výpis z jednotky	Zprávy o stavu servo motorů

4.2 Zpráva pro řídicí jednotku přestavníků

Komunikační zpráva je rozdělena na osm bytů. Tato zpráva je zobrazena v Tab. 4.6
Zpráva pro nastavení řídicí jednotky

Tab. 4.6 Zpráva pro nastavení řídicí jednotky

ID	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
----	--------	--------	--------	--------	--------	--------	--------	--------

4.2.1 Seznam funkcí řídicí jednotky přestavníků (výhybkového kontroléru)

V této kapitole jsou vypsány funkce, které řídicí jednotka přestavníků umí. Tyto funkce jsou vypsány v Tab. 4.7.

Tab. 4.7 Kompletní seznam funkcí

	DATA	FUNKCE	Kapitola
Data 0	0x00 – 0xB5	Řízení servo motorů (manuální řízení)	4.2.2
	0xB6	Resetování jednotky	0
	0xB7	Nastavení času mezi otáčením servo motoru	4.2.4
	0xB8	Zpětná vazba na vyžádání	0
	0xB9	Nastavení dorazů	0
	0xBA	Výběr a přehození výhybek	4.2.7
	0xBB	Nastavení čísla jednotky	4.2.8

4.2.2 Řízení servomotorů (manuální řízení)

Manuální řízení není určené pro běžný chod řídicí jednotky přestavníků. Je dobré pro servisního technika nebo osobu, která je pověřená testováním přestavníků (servomotorů).
Příklad: Chci vyměnit servomotor a chci ověřit jeho provoz. Pak použiju manuální řízení. Určitě se najdou i další příklady využití tohoto řízení. V Tab. 4.8 je tabulka, která ukazuje nastavení pro manuální řízení servomotorů.

Výchozí nastavení servomotorů je vlevo 0x00 což znamená 0° natočení servomotorů. Hodnoty od 0x00 – 0xB4 jsou určeny k natočení servomotorů od 0° – 180°. Pro vypnutí servomotoru musíme zadat hodnotu 0xB5.

Tab. 4.8 Řízení jednotlivých servomotorů

	DATA	FUNKCE
Data 0	0x00 – 0xB4	Natočení serva 1
	0xB5	Vypnutí serva 1
Data 1	0x00 – 0xB4	Natočení serva 2
	0xB5	Vypnutí serva 2
Data 2	0x00 – 0xB4	Natočení serva 3
	0xB5	Vypnutí serva 3
Data 3	0x00 – 0xB4	Natočení serva 4
	0xB5	Vypnutí serva 4
Data 4	0x00 – 0xB4	Natočení serva 5
	0xB5	Vypnutí serva 5
Data 5	0x00 – 0xB4	Natočení serva 6
	0xB5	Vypnutí serva 6
Data 6	0x00 – 0xB4	Natočení serva 7
	0xB5	Vypnutí serva 7
Data 7	0x00 – 0xB4	Natočení serva 8
	0xB5	Vypnutí serva 8

4.2.3 Resetování řídicí jednotky přestavníků

Pokud dojde k závažné chybě na řídicí jednotce přestavníků, je k dispozici restartování řídicí jednotky (restartování mikrokontroléru). V mikrokontroléru je Watchdog timer (časovač). Tento Watchdog timer se periodicky resetuje, když jednotka bezchybně komunikuje s nadřazeným systémem. Časovač hlídá jednotlivé průběhy programu, pokud se změni hodnota průběhu, pak se okamžitě restartuje mikrokontrolér. Při chybném nastavení hodnoty je chyba signalizována příslušnou barvou v kapitole 4.4. Zpráva pro restartování řídicí jednotky přestavníku je v Tab. 4.9.

Tab. 4.9 Resetování jednotky

Data	Číslo [Hex]	Funkce
Data 0	0xB6	Resetování jednotky
Data 1	0x01	Resetování potvrzeno
Data 2	0x00	Nevyužito
Data 3	0x00	Nevyužito
Data 4	0x00	Nevyužito
Data 5	0x00	Nevyužito
Data 6	0x00	Nevyužito
Data 7	0x00	Nevyužito

4.2.4 Nastavení časové prodlevy mezi otáčením servomotoru

Výchozí nastavení časové prodlevy mezi otáčením servomotoru je 0 ms. Toto nastavení moc nedoporučuji, protože dochází k velkému odběru proudu, kdybychom chtěli řídit najednou třeba všech osm servomotorů. Z tohoto důvodu doporučuji nastavení časové prodlevy mezi otáčením servomotoru asi tak jednu sekundu (pro jednu sekundu zadávejte 0x64). Dále nastavení času je od 0 do 2500 ms, při překročení hodnoty je chyba signalizována příslušnou barvou v kapitole 4.4. V Tab. 4.10 je ukázka, jak postupovat při nastavení času mezi servomotory.

Jak vypočítat nastavení času mezi otáčením servomotoru? Mikrokontrolér je nastaven na 10 ms. Pro zjištění hexadecimálního čísla Y musíme nejprve dosadit do vzorečku v desítkové soustavě a potom číslo X převedeme na hexadecimální číslo.

$$X = \frac{\text{čas (prodleva)}}{10} = \text{například pro } 1000 \text{ ms (1 sekunda)} = \frac{1000}{10} = 100$$

Potom co známe $X = 100$ dekadickou hodnotu převedeme na hodnotu hexadecimální $Y = 0x64$.

Tab. 4.10 Nastavení času mezi otáčením servomotoru

Data	Číslo [Hex]	Funkce
Data 0	0xB7	Nastavení času mezi otáčením servo motoru
Data 1	0x00	Čas 0 ms (otočí se všechny servo motory najednou)
	0x64	Čas 1000 ms (otočí se servo motory 1000 ms po sobě)
	0xFA	Čas 2500 ms (otočí se servo motory 2500 ms po sobě)
Data 2	0x00	Nevyužito
Data 3	0x00	Nevyužito
Data 4	0x00	Nevyužito
Data 5	0x00	Nevyužito
Data 6	0x00	Nevyužito
Data 7	0x00	Nevyužito

4.2.5 Zpětná vazba na vyžádání

Tato funkce nám zajistí přehled, co se děje na kolejišti s řídicí jednotkou přestavníků. Když zvolíme zpětnou vazbu, můžeme snadno zjistit, jakou má přestavník pozici. Pro zjištění pozice přestavníku jsem vytvořil dvě zpětné vazby v podobě zprávy. V Tab. 4.11 je způsob volání funkce. V sloupci Data, řádek Data 1 je nastavení výběru zprávy. V Tab. 4.12 je popsáno o jaký typ zprávy se jedná při zadání 0x01 a 0x02. Při chybném nastavení hodnoty je chyba signalizována příslušnou barvou v kapitole 4.4. V kapitole 0 je detailní rozbor zpětné zprávy.

Tab. 4.11 Zpětná vazba pro zjištění pozice servomotorů

Data	Číslo [Hex]	Funkce
Data 0	0xB8	Zpětná vazba na vyžádání
Data 1	0x01 – 0x02	Výběr typu zprávy (viz. Tab. 10)
Data 2	0x00	Nevyužito
Data 3	0x00	Nevyužito
Data 4	0x00	Nevyužito
Data 5	0x00	Nevyužito
Data 6	0x00	Nevyužito
Data 7	0x00	Nevyužito

Tab. 4.12 Popis funkce pro vybrání typu zprávy

	DATA	FUNKCE
Data 1	0x01	Pozice servo motoru vpravo, vlevo a mezi
Data 1	0x02	Pozice servo motorů ve stupních

4.2.6 Nastavení dorazů

Tato funkce slouží k nastavení dorazů přestavníku vlevo i vpravo. Hodnota se dá nastavit od 0° do 180°. Následující Tab. 4.13 zpřehledňuje toto nastavení. Hodnota nastaveného dorazu se uloží (viz. Kapitola 0).

Tab. 4.13 Nastavení dorazů doleva a doprava

Data	Číslo [Hex]	Funkce
Data 0	0xB9	Nastavení dorazů
Data 1	0x00 – 0x07	Výběr serva
Data 2	0x00 – 0xB4	Hodnota levého dorazu
Data 3	0x00 – 0xB4	Hodnota pravého dorazu
Data 4	0x00	Nevyužito
Data 5	0x00	Nevyužito
Data 6	0x00	Nevyužito
Data 7	0x00	Nevyužito

4.2.6.1 Uložení dorazů

V Tab. 4.14 je znázorněn způsob uložení do paměti EEPROM.

Tab. 4.14 Uložení

Data 1 (Servo motor)	Data 2 (Levý doraz)	Data 3 (Pravý doraz)
0x00 (1. servo motor)	0x00-0xB4	0x00-0xB4
0x01 (2. servo motor)	0x00-0xB4	0x00-0xB4
0x02 (3. servo motor)	0x00-0xB4	0x00-0xB4
0x03 (4. servo motor)	0x00-0xB4	0x00-0xB4
0x04 (5. servo motor)	0x00-0xB4	0x00-0xB4
0x05 (6. servo motor)	0x00-0xB4	0x00-0xB4
0x06 (7. servo motor)	0x00-0xB4	0x00-0xB4
0x07 (8. servo motor)	0x00-0xB4	0x00-0xB4

4.2.7 Výběr a přehození přestavníků (výhybek)

Budeme-li chtít přehodit přestavník musíme mít nastavené dorazy (viz. Kapitola 0), nebo můžeme zvolit manuální řízení (viz. Kapitola 4.2.2). Pokud máme nastavené dorazy můžeme přehodit vybrat a přehodit přestavník. V Tab. 4.15 je funkce na přehazování přestavníků. Tím jenom zvolíme, že budeme přehazovat přestavníky. Potom musíme dát pokyn k přehození a tento pokyn je popsán v kapitole 0.

Tab. 4.15 Výběr přestavníků

Data	Číslo [Hex]	Funkce
Data 0	0xBA	Přehod' výhybku
Data 1	0x00 – 0xFF	Přehození výhybky
Data 2	0x00 – 0xFF	Výběr výhybky
Data 3	0x00	Nevyužito
Data 4	0x00	Nevyužito
Data 5	0x00	Nevyužito
Data 6	0x00	Nevyužito
Data 7	0x00	Nevyužito

4.2.7.1 Přehození přestavníků (výhybek)

Pokud chceme přehodit přestavník doleva musíme zadat logickou jedničku (přehození doleva). Pro logickou nulu tedy platí přehození vpravo. V Tab. 4.16 je ukázka přehazování vlevo a vpravo. Osmý přestavník (Servo 8) je přehozen vlevo, za to sedmý přestavník (Servo 7) je přehozen vpravo. Přestavník se přesune doleva nebo doprava na pozici příslušného dorazu (viz. Kapitola 0). V kapitole 0 si ukážeme, jak vybrat libovolné přestavníky pro přehození pozice.

Tab. 4.16 Přehození doleva či doprava

Přehození serva	8	7	6	5	4	3	2	1
Data 1 [Bin]	1	0	0	1	0	0	0	1

Pro uvedený příklad je třeba zapsat hodnotu do Data 1 (viz. Tab. 4.15) v hexadecimálním tvaru. V tomto případě by šlo o hodnotu 1001 0001 v binární (dvojkové soustavě) po převedení by hodnota byla 0x91 v hexadecimálním tvaru (šestnáctková soustava).

4.2.7.2 Výběr přestavníku (výhybky)

Na jedné jednotce je osm přestavníků. Pokud chceme vybrat příslušný přestavník stačí jej vybrat podle Tab. 4.17. Například pro aktivaci přestavníku číslo osm (Servo 8) zadáme logickou jedničku a pro ignorování přestavníku sedm (Servo 7) zadáme logickou nulu.

Tab. 4.17 Výběr přestavníku k přehození

Výběr serva	8	7	6	5	4	3	2	1
Data 2 [Bin]	1	0	0	1	0	0	0	1

Pro uvedený příklad je třeba zapsat hodnotu do Data 2 (viz. Tab. 4.15) v hexadecimálním tvaru. V tomto případě by šlo o hodnotu 1001 0001 v binární (dvojkové) soustavě po převedení by hodnota byla 0x91 v hexadecimálním tvaru (šestnáctková soustava).

4.2.8 Nastavení čísla řídicí jednotky přestavníků

Každá řídicí jednotka přestavníků má svoje identifikační číslo, které je uloženo v paměti mikrokontroléru EEPROM. V počátečním nastavení má řídicí jednotka přednastavené číslo 1. Posledních sedm bitů ID umožňuje mít až 128 jednotek (viz. Tab. 4.3). Pokud by došlo k nechtěnému změnění čísla jednotky je zavedena tzv. kontrola, v podobě bitové negace čísla řídicí jednotky, která musí být zapsaná do Data 2. Pro změnění čísla řídicí jednotky je potřeba zapisovat do Data 1 číslo, které má řídicí jednotka a do Data 2 zapsat negaci tohoto čísla a poté se provede změna čísla řídicí jednotky. V Tab. 4.18 je nastavení popsáno. Při chybném nastavení hodnoty je chyba signalizována příslušnou barvou v kapitole 4.4.

Tab. 4.18 Nastavení čísla jednotky

Data	Číslo [Hex]	Funkce
Data 0	0xBB	Nastavení čísla jednotky
Data 1	0x00	Číslo jednotky: 0
	0x0A	Číslo jednotky: 10
	0x7F	Číslo jednotky: 127
Data 2	0xFF	Negace čísla 0
	0xF5	Negace čísla 10
	0x80	Negace čísla 127
Data 3	0x00	Nevyužito
Data 4	0x00	Nevyužito
Data 5	0x00	Nevyužito
Data 6	0x00	Nevyužito
Data 7	0x00	Nevyužito

4.3 Zpráva o natočení servomotorů přestavnickového mikrokontroléru

ID	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
----	--------	--------	--------	--------	--------	--------	--------	--------

Struktura zprávy záleží na příkazu uvedeného v kapitole 0.

Tab. 4.19 Zpráva o natočení servomotorů vpravo, vlevo a mezi

Data pro 0x01	DATA	Význam
Data 0	0xF0 – 0xF2	Natočení serva 1
Data 1	0xF0 – 0xF2	Natočení serva 2
Data 2	0xF0 – 0xF2	Natočení serva 3
Data 3	0xF0 – 0xF2	Natočení serva 4
Data 4	0xF0 – 0xF2	Natočení serva 5
Data 5	0xF0 – 0xF2	Natočení serva 6
Data 6	0xF0 – 0xF2	Natočení serva 7
Data 7	0xF0 – 0xF2	Natočení serva 8

Tab. 4.20 Zpráva o natočení servomotorů ve stupních

Data pro 0x02	DATA	Význam
Data 0	0x00 – 0xB4	Natočení serva 1
Data 1	0x00 – 0xB4	Natočení serva 2
Data 2	0x00 – 0xB4	Natočení serva 3
Data 3	0x00 – 0xB4	Natočení serva 4
Data 4	0x00 – 0xB4	Natočení serva 5
Data 5	0x00 – 0xB4	Natočení serva 6
Data 6	0x00 – 0xB4	Natočení serva 7
Data 7	0x00 – 0xB4	Natočení serva 8

Pro funkci 0x01

Hodnoty se porovnají s uloženou aktuální hodnotou natočení. Následně se zjistí, jestli souhlasí hodnota s dorazem vpravo nebo vlevo, poté se vypíše v podobě 0xF0 nebo v 0xF1. Když nesouhlasí hodnota ani s jednou hodnotou, pak se vypíše 0xF2. Poté je tato hodnota chápána jako výhybka mezi dorazy. Tabulka znázorňuje zpětnou zprávu o pozici výhybky (viz. Tab. 4.21).

Tab. 4.21 Pro funkci 0x01

Zpětná zpráva [Hex]	Význam zprávy
0xF0	Výhybka na dorazu vpravo
0xF1	Výhybka na dorazu vlevo
0xF2	Výhybka mezi dorazy

Pro funkci 0x02

Pozice servo motorů se vypisují ve stupních. Stupně natočení je třeba převést z šestnáctkové soustavy na dekadickou. Vypisují se v rozmezí 0x00 – 0xB4 odpovídá stupňům natočení od 0° do 180°. V následující tabulce je ukázka převodu (viz. Tab. 4.22).

Tab. 4.22 Pro funkci 0x02

Zpětná zpráva [Hex]	Převod na stupně [°]
0x00	0°
0x5A	90°
0x64	100°
0xB4	180°

4.4 Chybové hlášení přestavnickového mikrokontroléru

Pokud dojde k nějaké chybě je třeba signalizovat chybu. Signalizace chyby je realizována přes RGB LED. Pro chybu je hlášení značeno 0x01, pokud nenastala žádná chyba, pak je hlášení značeno 0x00. V Tab. 4.23 je popsáno ukládání do dat chybového hlášení.

Tab. 4.23 Chybové hlášení

Data chybového hlášení	DATA	Význam chyby
Data 0	0xFF	Začátek chybové zprávy
Data 1	0x00 – 0x01	Chybné volání funkce
Data 2	0x00 – 0x01	Chyba při nastavení servo motorů
Data 3	0x00 – 0x01	Chyba při HARD resetu
Data 4	0x00 – 0x01	Chybné nastavení času mezi otáčením servo motoru
Data 5	0x00 – 0x01	Chyba při zpětné vazbě
Data 6	0x00 – 0x01	Chyba při nastavení čísla jednotky
Data 7	0xFF	Konec chybové zprávy

V Tab. 4.24 je signalizace RGB LED. Jak můžeme z tabulky vidět, tak zelená barva znamená, že nedošlo k žádné chybě. Pokud bude zadaný příkaz špatně, než bylo v předchozích řádcích komunikačního protokolu popsáno, pak bude svítit modrá barva. Barva červená signalizuje chybu při HARD resetu a program bude zastaven. Žlutá barva je pro chybné nastavení přestavníku např. špatná hodnota (nad 0xB5). Fialová je pro špatně zadanou zpětnou vazbu v Data 1. Azurová je pro chybné nastavení prodlevy mezi otáčením přestavníků (povolená hodnota je od 0 až 2500 ms). Bílá barva je pro špatné nastavení čísla řídicí jednotky.

Tab. 4.24 Signalizace RGB LED

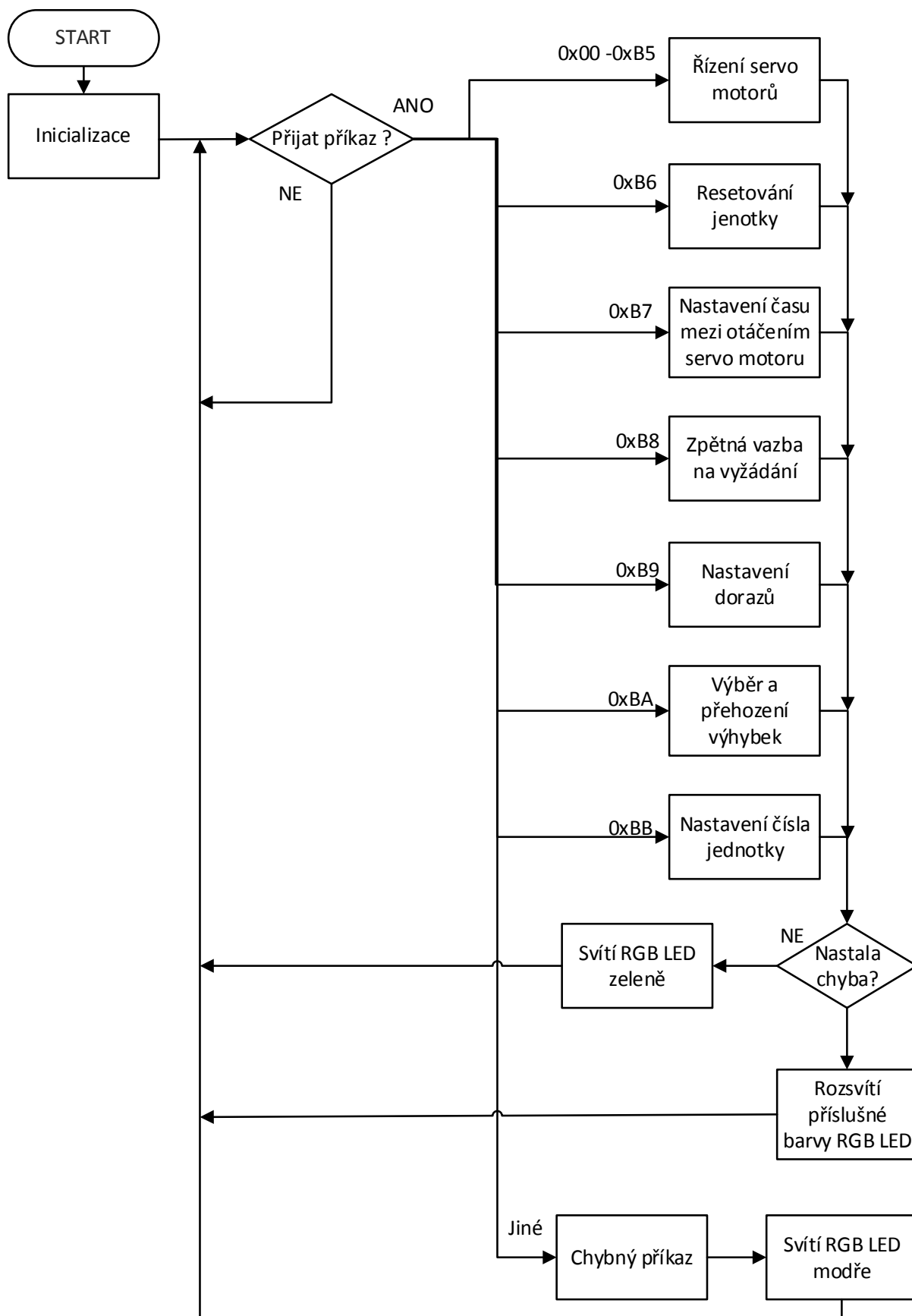
Barva RGB	Význam chyby
Zelená	Všechno v pořádku
Červená	Chyba při HARD resetu
Modrá	Chybné volání funkce
Žlutá	Chyba při nastavení servomotorů
Fialová	Chyba při zpětné vazbě
Azurová	Chybné nastavení času mezi otáčením servomotoru
Bílá	Chyba při nastavení čísla jednotky

5 Software

Program je napsán v jazyce C v počítačový programu CodeWarrior. Přes aplikaci Processor Expert, kterou disponuje CodeWarrior, proběhla inicializace periférií mikrokontroléru.

5.1 Vývojový diagram

Funkci celého programu znázorňuje vývojový diagram, který je na Obr. 5.1. Při zapnutí programu se načtou deklarace proměnných a nastavení výchozích hodnot. Poté následuje inicializace podprogramů. Jak můžeme vidět z vývojového diagramu, tak program běží v nekonečné smyčce. V této smyčce je funkce switch. Tato funkce je popsána ve vývojovém diagramu jako „Přijít příkaz“, zde dochází k rozhodování, když se rozhodneme zadávat nějakou funkci nebo se nic dít nebude. Tyto funkce jsou popsány v komunikačním protokolu kapitola 4.2.1. Pokud se nic dít nebude, pak řídicí jednotka nic neudělá, ale program dál běží v nekonečné smyčce a čeká na zadání příkazu. Pokud je ale zadán příkaz, dojde k rozhodnutí, jestli je příkaz dobře zadán nebo nikoliv. Pokud není dobře zadán příkaz, pak řídicí jednotka vyhodnotí tento příkaz jako chybný a zobrazí chybu příslušnou signalizací RGB LED (v tuhle chvíli modrou barvou). Pakliže je příkaz správný, poté řídicí jednotka povolí vstup do požadované funkce a bude kontrolovat správné zadání parametru. Pokud je chybně zadán parametr pro danou funkci, pak řídicí jednotka opět vyhodnotí chybné zadání parametru a zobrazí se signalizace RGB LED jako chybná. Tyto signalizace chyb jsou vypsány v komunikačním protokolu v kapitole 4.4. Pokud je správně zadán parametr funkce, pak požadovanou funkci řídicí jednotka považuje jako za správně zadanou a následuje vykonání dané činnosti. Správný běh programu hlídá časovač Watchdog, který je popsán v kapitole 2.2. Dále správný chod programu je signalizován RGB LED barvou zeleně. Pokud dojde k nějaký chybě, tak program vyskočí na začátek „Přijít příkaz“ a program poběží v posledním správné nastavení. Pro změnu posledního nastavení je třeba zopakovat celou akci a zadat správné volání funkce a parametry funkce.



Obr. 5.1 Vývojový diagram pro řídicí jednotku přestavníků

6 Závěr

Tato práce se zabývala vytvořením řídicí jednotky přestavníků pro modelovou železnici, která dokáže komunikovat po sběrnici CAN. Jako řídicí prvky byly použity stejnosměrné servomotory s otáčením od 0° do 180°. Práce zahrnuje funkci modelového kolejiště přes sběrnici CAN. Dále byl vytvořen návrh desky plošných spojů a rozmístění součástek pro řídicí jednotku přestavníků. Na řídicí jednotku se vejde osm servomotorů, které ovládají přestavníky pro modelové kolejiště. Po vytvoření desky plošných spojů byl vytvořen detailně kompletní komunikační protokol, který popisuje funkce řídicí jednotky. Poslední část byla věnována programování v jazyce C.

V této práci se podařilo navrhnout kompletní hardware a postavit prototypovou desku, kterou jsem sám ručně osadil a zapájel. Deska byla testována, jestli komunikuje s CAN sběrnici, a to se povedlo. Zprávy byly přijímány i odesílány a následně plnila jednoduché funkce, které byli testovány na rozsvěcování RGB LED. Potom jsem udělal úpravy na desce pro lepší design a funkce desky. Dále jsem zkompletoval celý komunikační protokol, který by měl být nástrojem k pochopení fungování celé řídicí jednotky a měl by tak pomoci mým následníkům, kteří se budou zabývat vylepšením mé dosavadní práce.

V této práci se nepodařilo zkompletovat celý program pro řídicí jednotku přestavníků, protože bych potřeboval přístup k modelovému kolejišti. Přístup jsem neměl z důvodu pandemie, která vypukla v době, kdy jsem pracoval na této práci.

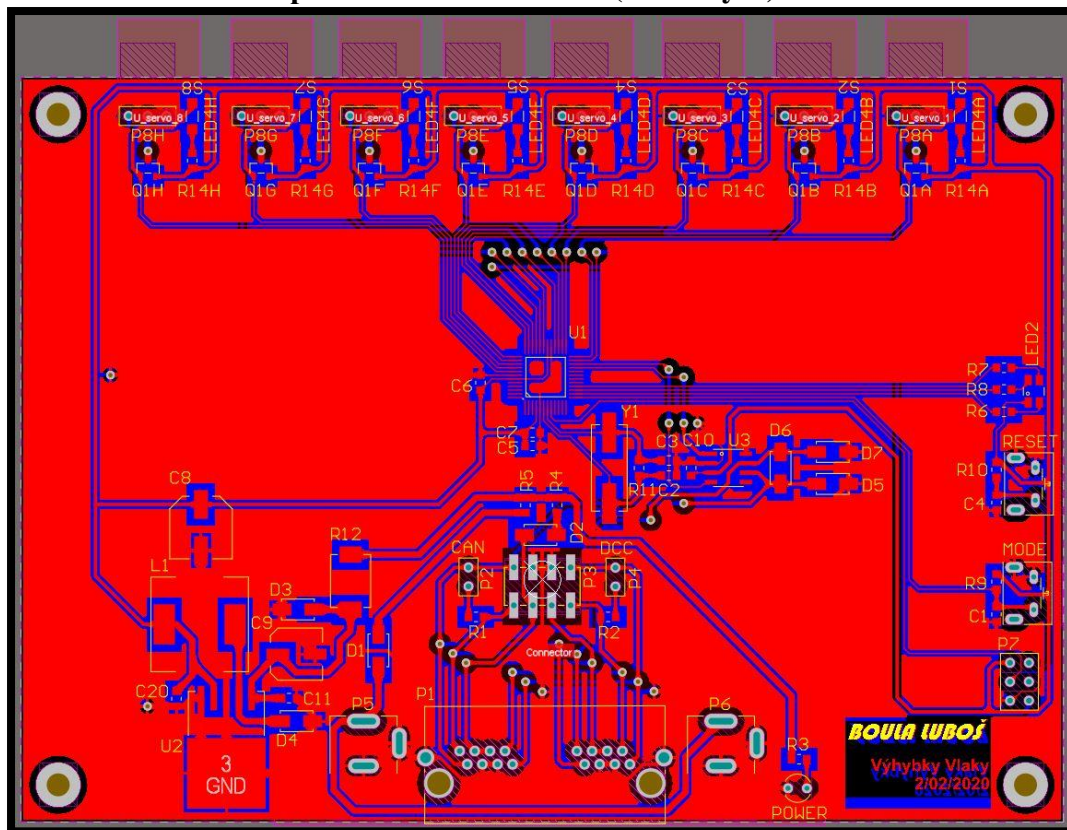
Tato práce mi zabrala spoustu času, ale myslím si, že se nakonec povedla a Vám čtenářům, se bude líbit a pomůže Vám k pochopení vašeho cíle.

Seznam literatury a informačních zdrojů

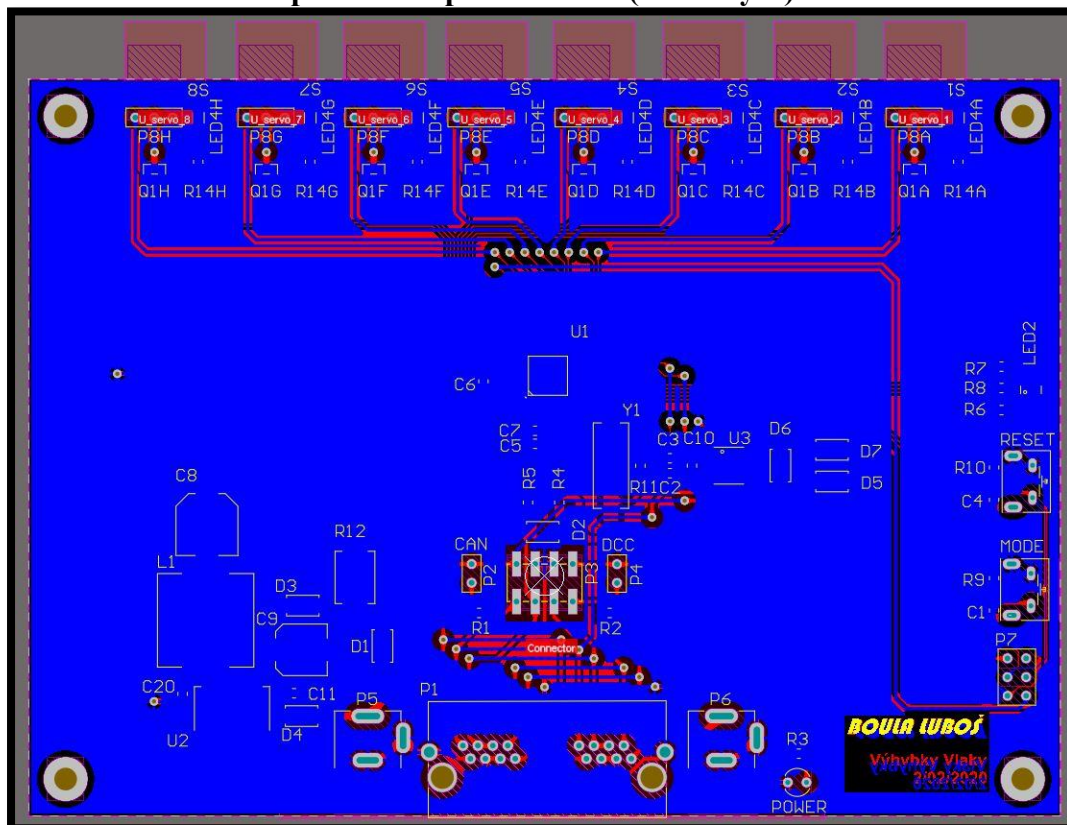
- [1] WEISSAR, Petr, Jiří ŽAHOUR a Ondřej LUFINKA. *Vlaky TT – Projekty FEL - Wiki* [online]. Plzeň, 2015, 11. dubna 2019 [cit. 2020-06-08]. Dostupné z: http://projekty.fel.zcu.cz/index.php/Vlaky_TT
- [2] MALENA, Ondřej. *Bakalářská práce: Řídicí jednotka kolejových úseků pro modelovou železnici* [online]. Plzeň, 2018 [cit. 2020-06-08]. Dostupné z: https://portal.zcu.cz/StagPortletsJSR168/PagesDispatcherServlet?pp_destElement=%23ssSouboryStudentuDivId_13318&pp_locale=cs&pp_reqType=render&pp_portlet=souboryStudentuPagesPortlet&pp_page=souboryStudentuDownloadPage&pp_nameSpace=G566744&soubidno=123064
- [3] *Bastlení - jak funguje modelářské servo* [online]. CZ, 2017 [cit. 2020-06-08]. Dostupné z: <http://vlastikd.webz.cz/bastl/serva.htm>
- [4] *Dsh.331-061.1.pdf* [online]. USA, 2004 [cit. 2020-06-08]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.331-061.1.pdf>
- [5] *High-speed CAN Transceiver - dsh.433-300.1.pdf* [online]. USA, 2003 [cit. 2020-06-08]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.433-300.1.pdf>
- [6] *MC9S08DZ60, MC9S08DZ48, MC9S08DZ32, MC9S08DZ16 - Data Sheet - MC9S08DZ60.pdf* [online]. USA, 2008 [cit. 2020-06-08]. Dostupné z: <https://www.nxp.com/docs/en/data-sheet/MC9S08DZ60.pdf>
- [7] ZVONARĚ, Filip. *Bakalářská práce: Generátor DCC signálu pro modelovou železnici* [online]. Plzeň, 2018 [cit. 2020-06-08]. Dostupné z: <http://projekty.fel.zcu.cz/images/f/f8/BP-zvonar.pdf>
- [8] KALČÍK, Martin. *Bakalářská práce* [online]. Plzeň, 2013 [cit. 2020-06-08]. Dostupné z: <https://dSPACE5.zcu.cz/bitstream/11025/8247/1/Sbernice%20CAN.pdf>

Přílohy

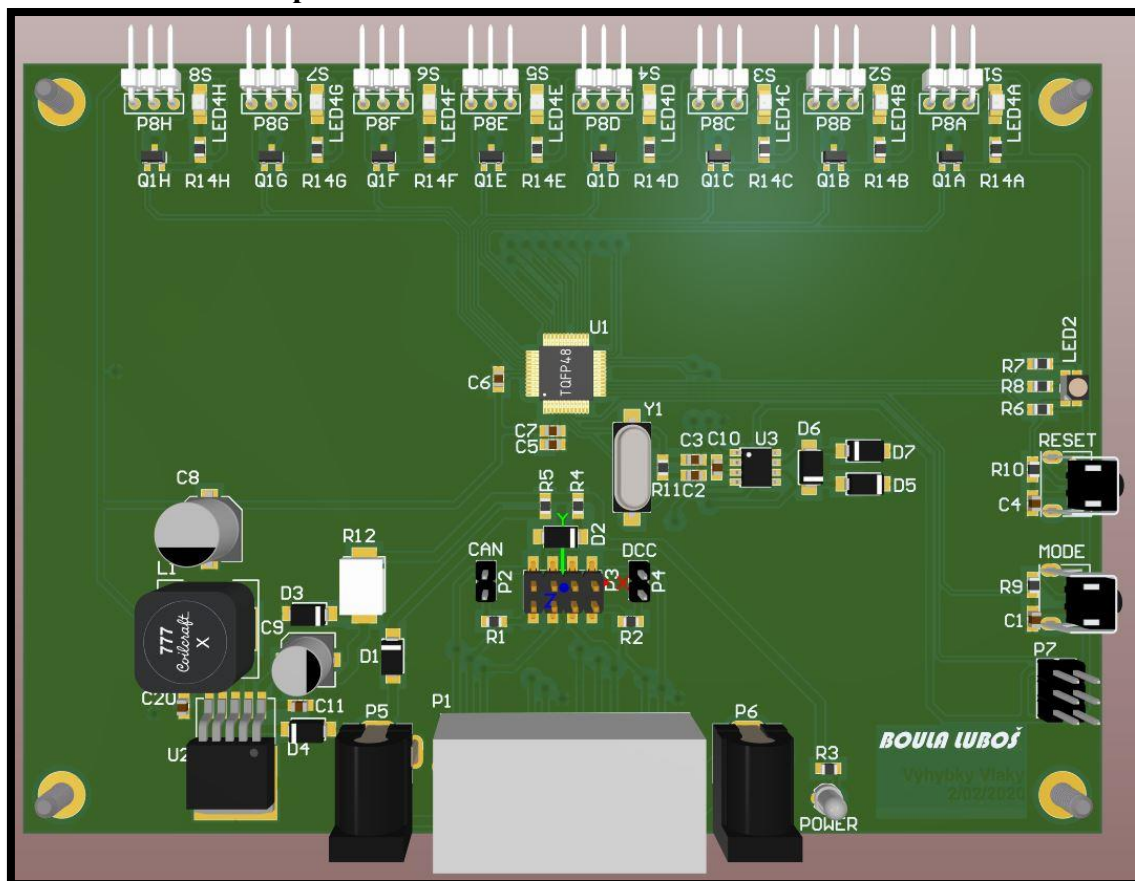
Obrázek 1.: DPS 2D pohled na horní stranu (TOP layer)



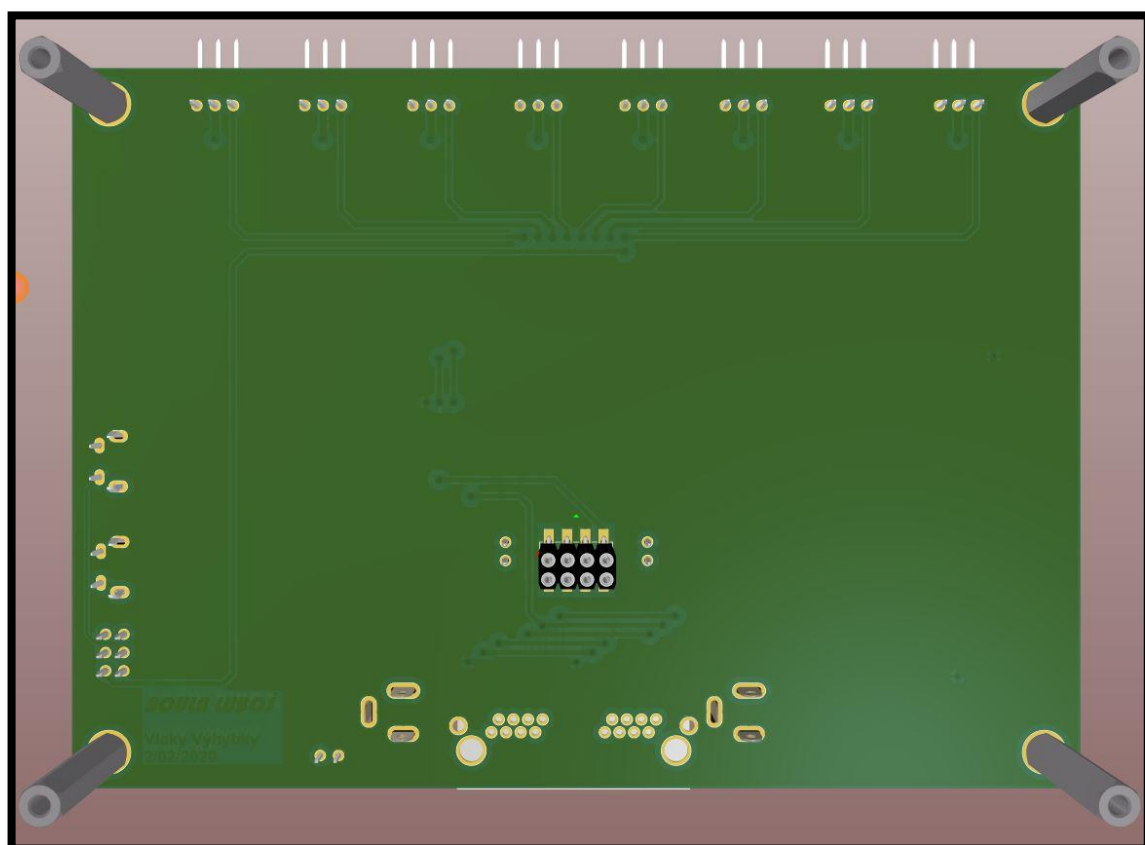
Obrázek 2.: DPS 2D pohled na spodní stranu (BOT layer)



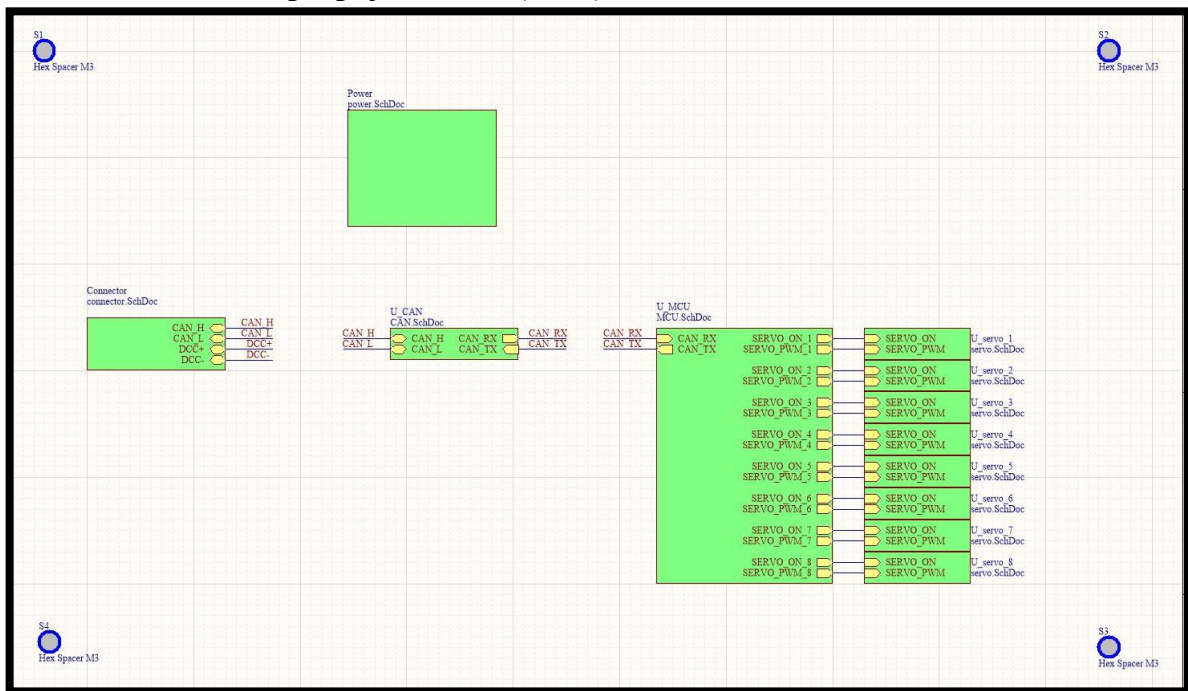
Obrázek 3.: DPS 3D pohled na horní stranu



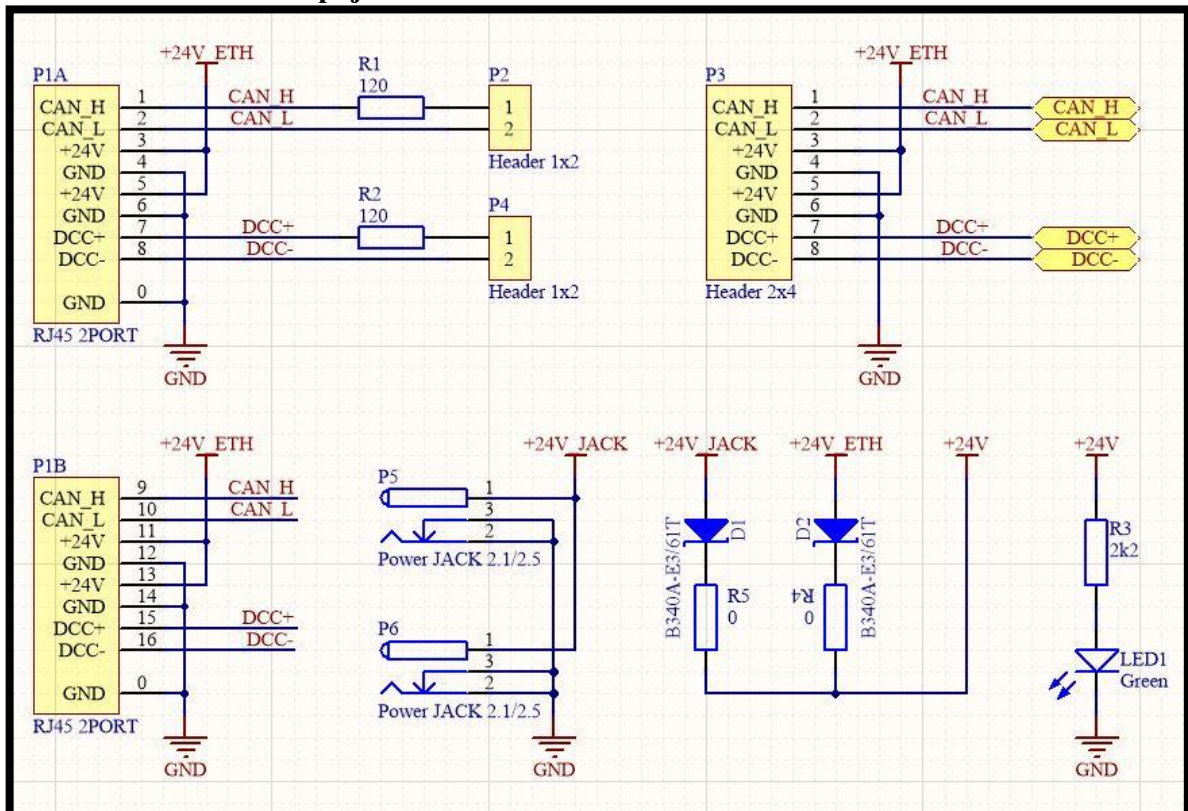
Obrázek 4.: DPS 3D pohled na spodní stranu



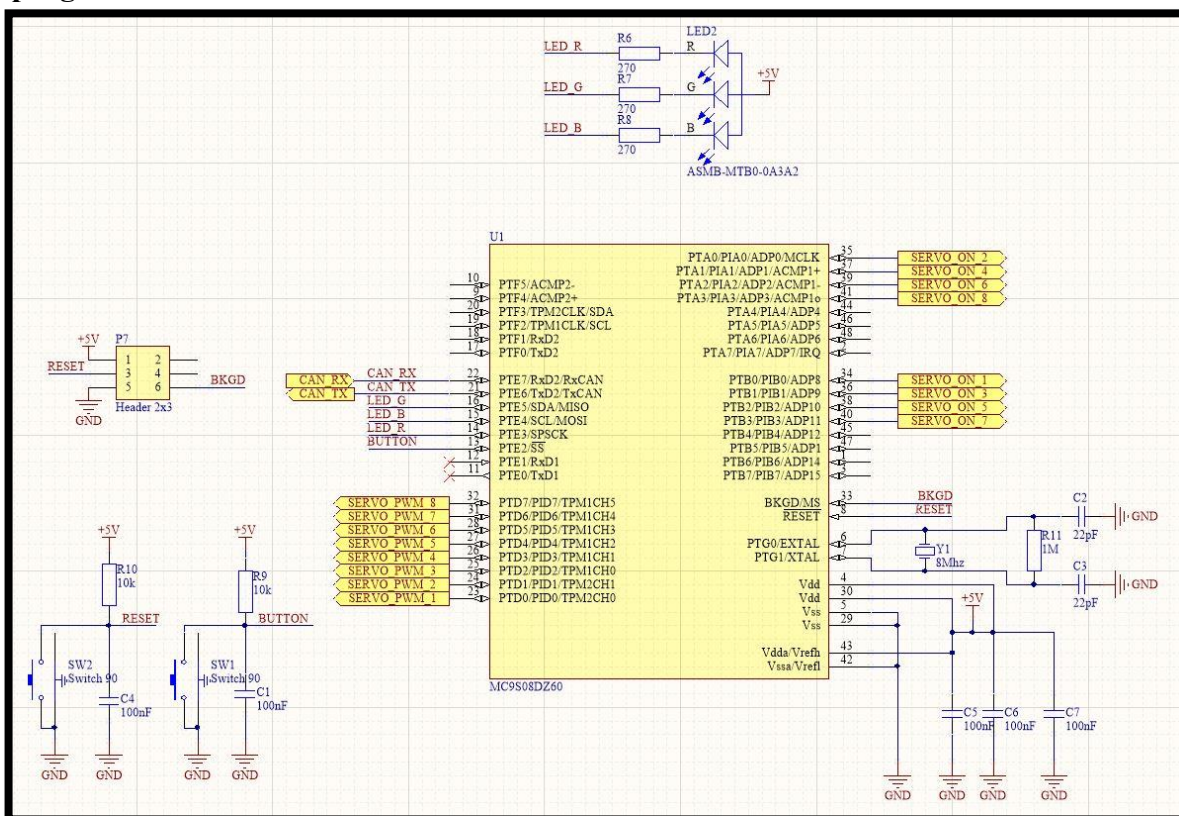
Obrázek 5.: Schéma propojení bloků (main)



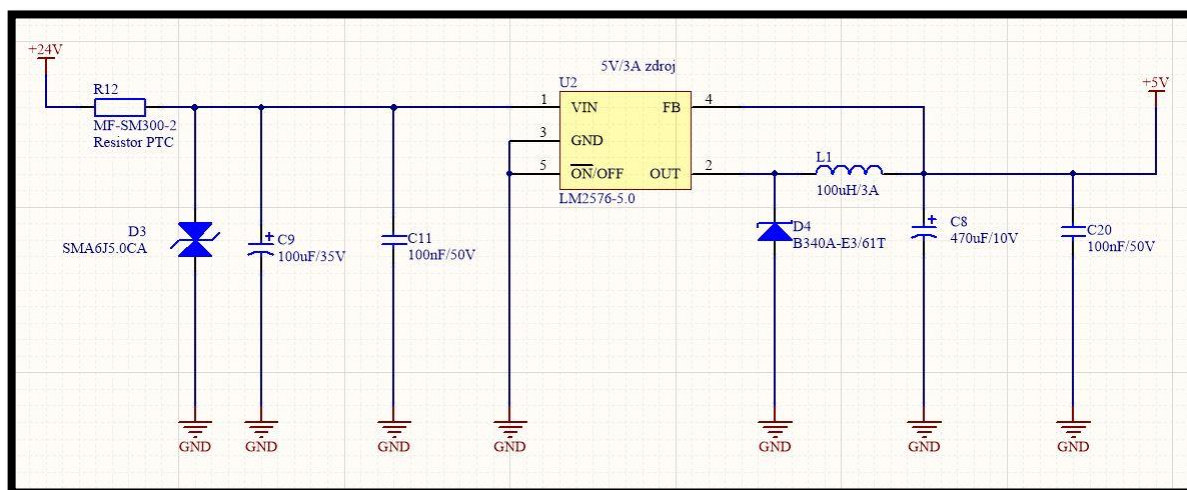
Obrázek 6.: Schéma zapojení konektorů



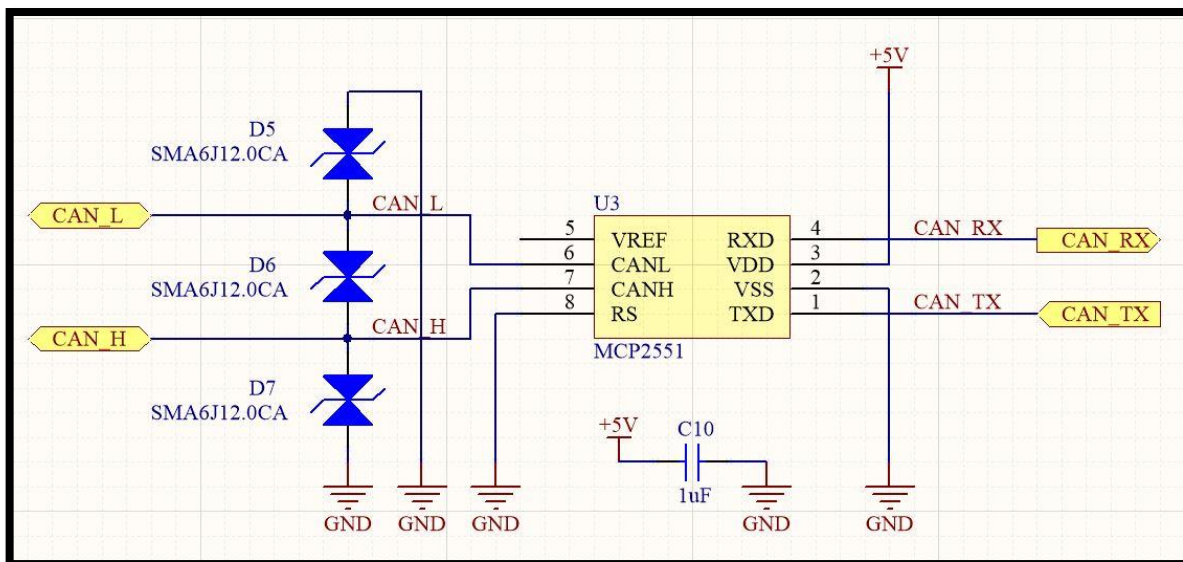
Obrázek 7.: Schéma zapojení mikrokontroléru, RGB LED, tlačítek a programovacího konektoru.



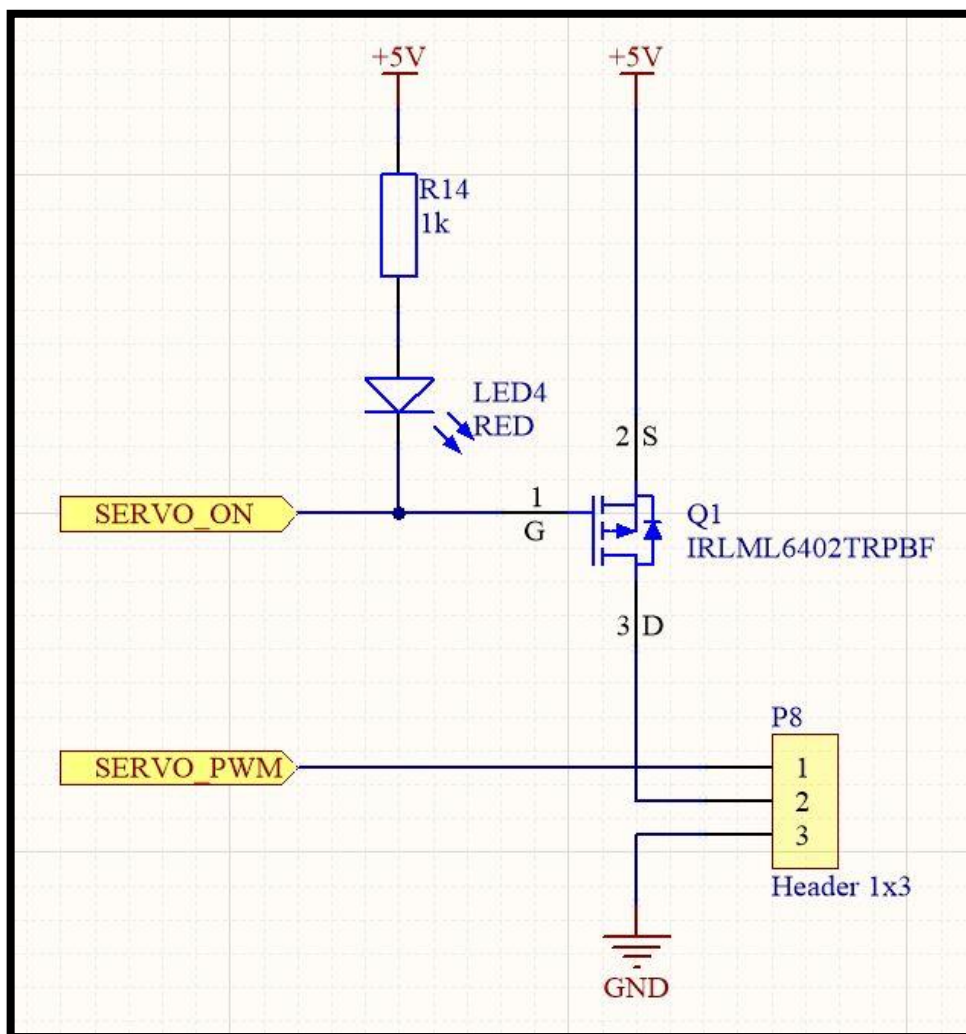
Obrázek 8.: Schéma zapojení spínaného zdroje LM2576.5.0



Obrázek 9.: Schéma zapojení CAN budiče



Obrázek 10.: Schéma zapojení pro servomotor



Obrázek 11.: Seznam součástek (BOM)

Comment	Description	Designator	Footprint	Libref	Quantity
100nF	Capacitor	C1, C4, C5, C6, C7	SMD_CAPACTOR_0805	Capacitor	5
22pF	Capacitor	C2, C3	SMD_CAPACTOR_0805	Capacitor	2
470uF/10V	Capacitor_POL	C8	SMD_CAPACTOR_E_8MM	Capacitor_POL	1
100uF/35V	Capacitor_POL	C9	SMD_CAPACTOR_D_6.3MM	Capacitor_POL	1
1uF	Capacitor	C10	SMD_CAPACTOR_0805	Capacitor	1
1000uF/50V	Capacitor	C11, C20	SMD_CAPACTOR_0805	Capacitor	2
B340A-E3/61T	Diode_Schottky	D1, D2, D4	SMD_DO-214AC	Diode_Schottky	3
SMA615.OCA	Transient Voltage Suppressor, Bidirectional	D3	SMD_DO-214AC	Transil	1
SMA6112.OCA	Transient Voltage Suppressor, Bidirectional	D5, D6, D7	SMD_DO-214AC	Transil	3
1000uH/3A	Inductor	L1	SMD_INDUCTOR_12.5MM	Inductor	1
Green	LED	LED1	THT_LED_3MM4	LED	1
ASMB-MT80-0A3A2	LED RCG8	LED2	SMD_LED_RCG8	LED RCG8	1
RED	LED RCG8	LED4, LED4B, LED4C, LED4D, LED4E, LED4F, LED4G, LED4H	SMD_LED_1206	LED	8
RA45 2PORT	RA45 - 2 port	P1	THT_RA45_2PORT	RA45_2PORT	1
Header 1x2	Header 1x2	P2, P4	THT_HEADER_100MIL_1X2_MALE	Header_1x2	2
Header 2x4	Header 2x4	P3	SMD_HEADER_100MIL_2X4_DOUBLESIDED	Header_2x4_Trains_Project	1
Power JACK 2.1/2.5	Low Voltage Power Supply Connector	P5, P6	THT_POWER_JACK_2.1/2.5	Power_JACK_2.1/2.5	2
Header 2x3	Header 2x3	P7	THT_HEADER_100MIL_1X3_MALE	Header_2x3	1
Header 1x3	Header 1x3	P8A, P8B, P8C, P8D, P8E, P8F, P8G, P8H	THT_HEADER_100MIL_1X3_MALE_90	Header_1x3	8
RJ45 RJ45 2TP9BF	Transistor MOSFET P	Q1A, Q1B, Q1C, Q1D, Q1E, Q1F, Q1G, Q1H	SMD_SOT-23	Transistor_MOSFET_P	2
120	Resistor	R1, R2	SMD_RESISTOR_0805	Resistor	2
2k2	Resistor	R3	SMD_RESISTOR_0805	Resistor	1
0	Resistor	R4, R5	SMD_RESISTOR_0805	Resistor	2
270	Resistor	R6, R7, R8	SMD_RESISTOR_0805	Resistor	3
10k	Resistor	R9, R10	SMD_RESISTOR_0805	Resistor	2
1M	Resistor	R11	SMD_RESISTOR_0805	Resistor	1
MF-SMB00-2	Resistor PTC	R12	SMD_RESISTOR_PTC_MF-SM	Resistor_PTC	1
1k	Resistor	R14A, R14B, R14C, R14D, R14E, R14F, R14G, R14H	SMD_RESISTOR_0805	Resistor	8
Hex Spacer M3	Hexagon metal spacer M3	S1, S2, S3, S4	THT_HEX_SPACER_M3L20	HEX_SPACER_M3	4
Switch 90	Tactile Switch Right Angle	SW1, SW2	THT_SWITCH_SPST_90	Switch_90	2
MCP3908D260	8xP 8 Bit Microcontroller, CAN	U1	SMD_LQFP-48	MCP3908D260	1
LM2576-5.0	Step Down Switching Regulator, Fixed, 4V-40V Input, 5V and 3A Output	U2	SMD_TO-263-5	LM2576-5.0	1
MCP2551	CAN Transceiver, 4.5 V - 5.5 V, 1 Mb/s	U3	SMD_SOIC-8	MCP2551	1
8MHz	Crystal Oscillator	V1	SMD_XTAL_HC-49	XTAL	1